ONLINE SHOPPING SYSTEM
FOR
ECART.LK (PVT) LTD

W. A. N. M. KUMARANATHUNGA

BIT Registration Number: R141180

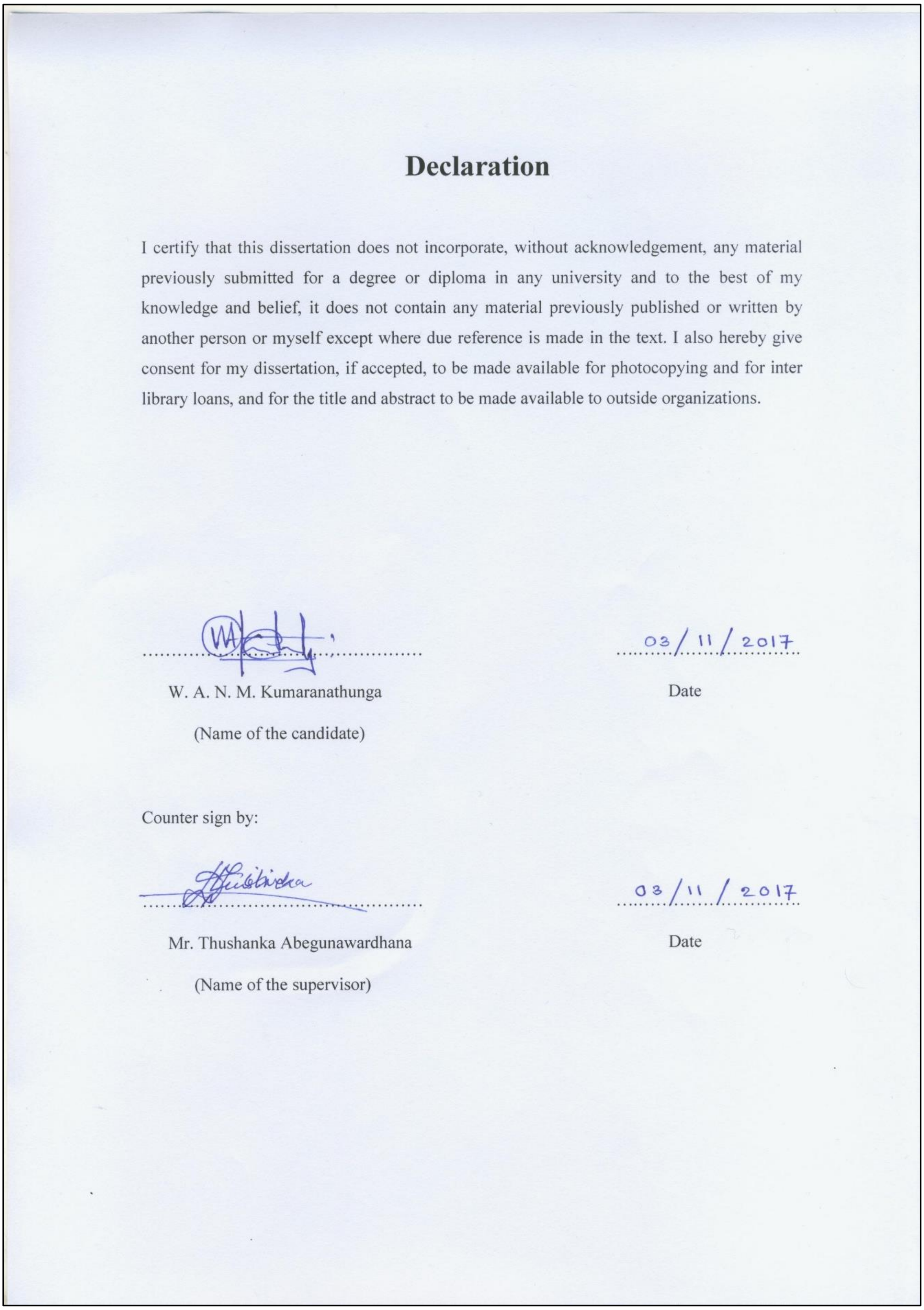
Index Number: 1411802

Name of Supervisor: Mr. W.T.S.D. Abeygunawardhane

November 2017

This dissertation is submitted in partial fulfilment of the requirement of the Degree of Bachelor of Information Technology (External) of the University of Colombo School of Computing

Declaration



Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for inter library loans, and for the title and abstract to be made available to outside organizations.

....	03/ 11 / 2017
W. A. N. M. Kumaranathunga	Date
(Name of the candidate)	

Counter sign by:

....	03 / 11 / 2017
Mr. Thushanka Abegunawardhana	Date
(Name of the supervisor)	

Abstract

Since the last decade, WWW has become a major factor in most of the fields. E-commerce or E-shopping has changed up to certain level due to popularity of internet, which delivers several advantages for the businesses as well as the customers' day today activities. Online shopping has made it easy for the customers' collaboration, communication as well as the valuable time. Customers can do their shopping activities in their fingertips by digitalizing the shopping environment.

The primary goal of an online shopping website and system is to sell goods and services using internet. This project provides facility to maintain the business which is system while facilitate the customers to go on shopping virtually in the internet which is website. Customers can fetch their desired items and purchase from the website. From the system perspective, business owners or employees in the company can control the business as well as the website.

This project was designed with Object Oriented Analysis and Design (OOAD) concept. Because of client had clear and stable idea about what they want from the system, project was undertaken with the iterative waterfall model. Resources that were used in this project were open source and JavaEE was selected as the main programming language in terms of development. Online shopping system which is a web based software solution was developed using JAVA backend and front end development use HTML, JavaScript and other several libraries. In order to build it using object oriented concept and MVC architecture, Java frameworks such as Struts, Spring and Hibernate were used. Tomcat Apache was used as application server and MySQL was the database.

The system was tested in multi-stage testing process by categorizing into two such as functional and non-functional testing which includes unit testing, integration testing and system testing, load testing, security testing and acceptance testing.

Acknowledgement

First, I would like to grant my gratitude to my beloved parents from the bottom of my heart for giving me the courage, strength, guidance, advice and blessings.

Then I express my thanks to the Bachelor of Information Technology (BIT) Coordinator of University of Colombo School of Computing (UCSC) and the project examination board of BIT for giving this great opportunity to do this project as a final project for BIT degree which help up to develop our carrier and programming development experience.

Then, my client Ecart.lk (PVT) Ltd. and the director of the company Mr. R. T. Jayasinghe who is a very busy person but reserved his valuable time so that he could give me for a discussion regarding the project.

I would like to show my greatest appreciation to my supervisor Mr. W.T.S.D Abeygunawardhana who is a senior employee in Openarc Systems for guiding and providing me with all the business knowledge and technical knowledge. Without his encouragement and guidance, it would be really difficult to carry out this project.

Openarc School of Business of Technology and its panel of dedicated lectures must receive my thanks for the knowledge, education and the wisdom I have gathered while being a student in there.

Finally, I take this opportunity to express my gratitude to my friends, colleagues and all the people those who help me in directly and indirectly to successfully complete this project.

Table of Content

List of Figures

List of Tables

List of Acronyms

CRUD	-	Create / Read / Update / Delete
CSS	-	Cascading Style Sheet
ERD	-	Entity Relationship Diagram
FK	-	Foreign key
HTML	-	Hyper Text Markup Language
IDE	-	Integrated Development Environment
JDK	-	Java Development Kit
JRE	-	Java Runtime Environment
JSP	-	Java Server Pages
J2EE	-	Java Enterprise Edition
MVC	-	Model View Controller
OO	-	Object Orient
OOD	-	Object Orient Design
ORM	-	Object Relation Mapping
PDF	-	Portable Document Format
PK	-	Primary key
UI	-	User Interface
UML	-	Unified Modeling Language
XML	-	Extensible Markup Language

Chapter 1 – Introduction

Ecart.lk (PVT) Ltd is a newly emerging and successful business which have about 10 staff members working now. They mainly sell goods and items like ornaments, electronic items, accessories, etc. They initially started their business by advertising in the Facebook. With the growth of their customer base and for marketing purposes they want to sell items through the internet via e-shopping.

Online shopping is very popular nowadays. Because many people love to purchase their items online and deliver them to their doorstep. Busy life style of modern society always like to do things in their fingertips.

Therefore Online Shopping system will be the best solution to overcome this problems.

1.1 Motivation for the project

E-Shopping is becoming a major trend in present day. People always prefer to do most of their day today activities without being tired and without wasting their time. Therefore Ecart.lk (PVT) Ltd, pays its attention toward the shopping through internet. It also helps to take the attention of the foreign buyers who would like to buy items through the e-shopping.

The benefit of the project is that the buyers can compare items with other brands and they have their full freedom to think about the goods and services which are getting through e-shopping. Seller (company) can target specific group of people whenever new items arrive to their shops. As an example if seller have sun glasses, they can promote those items in social media to take the attention of young generation.

1.2 Objectives and scope of the project

1.2.1 Objectives

- Enhance the range of customer base rather than customers who are visiting to shop.
- Give opportunity for customers to search items without wasting time and fuel.
- Managing the item deliveries which are ordering through the online shopping system.
- Provide facility to generate reports which are useful for management level.

1.2.2 Scope of the project

This online shopping system has online selling portal which is only done by the Ecart.lk (PVT) Ltd. Buyers of Ecart.lk can register into the system or if buyer is a onetime user or he/she has no interest in use our online shopping system again, then he/she can buy items as a guest user.

- System perspective
    - Administration level users
        - System administrator or company administrative users can manage stakeholders in the system who are working in the company (System users)
        - They can define access groups with the internal stakeholders and also access rights can be manage within the system.
        - Admin who is maintain the company website can be able to change website main banners and details of the company like location, description of the company, etc.
        - Admin users can manage products, product categories, items and also they can deicide discount prices, promotion items, minimum and maximum price ranges, offers, etc.
        - They can define shipping areas and appropriate shipping cost for the item deliveries.
    - Low level users
        - They can manage products, products categories and items with the authorization of managing level.
        - System allows to give reports like,
            - Monthly income.
            - Most selling items.
            - Most viewed items.
            - Daily transactions.
            - Best customers.
            - Available, sold out stocks.

        - System users can display the ordered items and drivers can assign to deliver the items if delivery location address is in the system defined area. Otherwise it is posted to the customer.
    - Customers
        - Customer or buyer cannot access to the system because customer is not considered as a seller or company employee.
- Website perspective
    - Customers
        - Customer can register into the system and registrations are authorize with the SMS notification or email link which is sent by the system.
        - Customer can add items to cart and purchase all of them as one order. If he/she wish to buy items later, then he/she can add those items to wish list for later usage.
        - They receive the SMS notification of purchase order, delivery startup and also estimated time for delivery by considering traffics and weather.
        - Money can be pay on delivery or online payments can be done through debit cards.
        - Registered customers can manage their user accounts and also see their order’s status, purchase histories
        - Customer able to search items according to their preference.
        - Customer can rate goods and items and also the delivery services after log into the website.

1.3 Structure of the Dissertation

Dissertation of the Online Shopping System provides the overall idea which provides the system. Structure of the dissertation can be summarize as follows.

Chapter 01 – Introduction

This chapter is one of most important chapter as it describe the motivation of the project as well as it gives the abstract idea of the developed system. Scope of the system and the objectives are described in this chapter.

Chapter 02 - Analysis

This chapter describes the business processes of the current system, requirement gathering techniques and also the functional and non-functional requirements of the developed system.

Chapter 03 - Design

Chapter 03 mainly includes the design of the system. UI design, architectural design and database design are clearly describe in this section with diagrams.

Chapter 04 - Implementation

Implementation of the system is described in chapter 4. Programming language, development tools, basic source codes, software and hardware environment used for implementing the system are described in this chapter.

Chapter 05 – Evaluation

This chapter describes about the testing procedures used in order to minimize the bugs and errors of the system.

Chapter 06 - Conclusion

The final chapter discusses about the critical evaluation, lessons learnt and future improvements.

Chapter 2 – Analysis

2.1 Introduction

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. [1]

2.2 Existing system

Ecart.lk Company has been doing their business activities in small scale while maintaining all the data in excel documents. They do not have a show room or shopping place. They do their advertising mainly in the social media like Facebook.

When a customer placed an order after going referring the Facebook advertisements, he / she call the contact number of the Ecart.lk or message the item codes with relevant details such as name, mobile number, quantity and delivery address.

Then the employees of the Ecart.lk enter all the details in to appropriate excel documents. Further, company reviews the order placed and call back to the customer to confirm the order. If the customer confirms the order, then order delivers to the relevant addresses of the customer. All payments are done in a traditional way which is cash on delivery.

All transaction records of the orders as well as the purchased items to the company are maintained in the excel documents. Available inventory, customer details and also reports are used to generated manually through that excel sheets.

2.3 Problems in the existing system

One of the main problem in the system is all the data records and logs are maintained in the excel sheets which are stored in a personal computers. Integrity of the collected data, confidentiality of the customers’ details and the data redundancy are main problems with regard to data maintenance.

The way of presenting the goods and items available in the company is not in proper manner. Customer can go through the items which are only posted / shared by the company. Customer cannot search for desired items or unable to compare with other similar items.

The process of order placement is time consuming and sometimes the verification of the customer is really difficult to address. Generating summary of the daily routine as well as monthly reports are hard for the company with document based system, unorganized system. Use case of the old system can be shown in the following Figure 2.1.

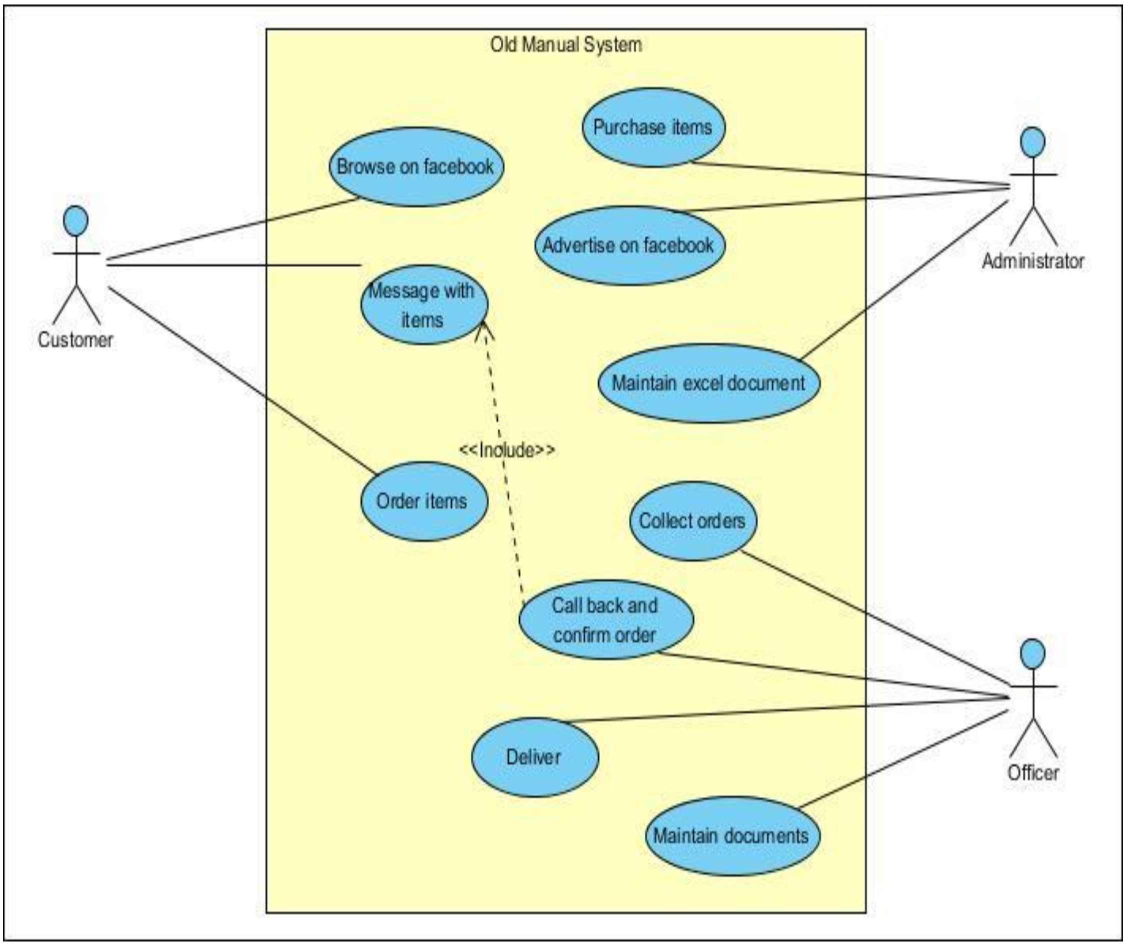


Figure 2.1: Use case for old manual system

In order to overcome almost all the issues in the existing system ‘Online Shopping System’ has become the best solution.

2.4 Existing similar products

2.4.1 Ebay.com

Ebay is a worldwide online shopping website where customers are allowed to buy and sell items via internet. Sample screen is shown below by Figure 2.2.

Features

- Facility to search with any keyword.
- Can compare items.
- Able to maintain a shopping cart.
- Provides facility to do monetary transactions online

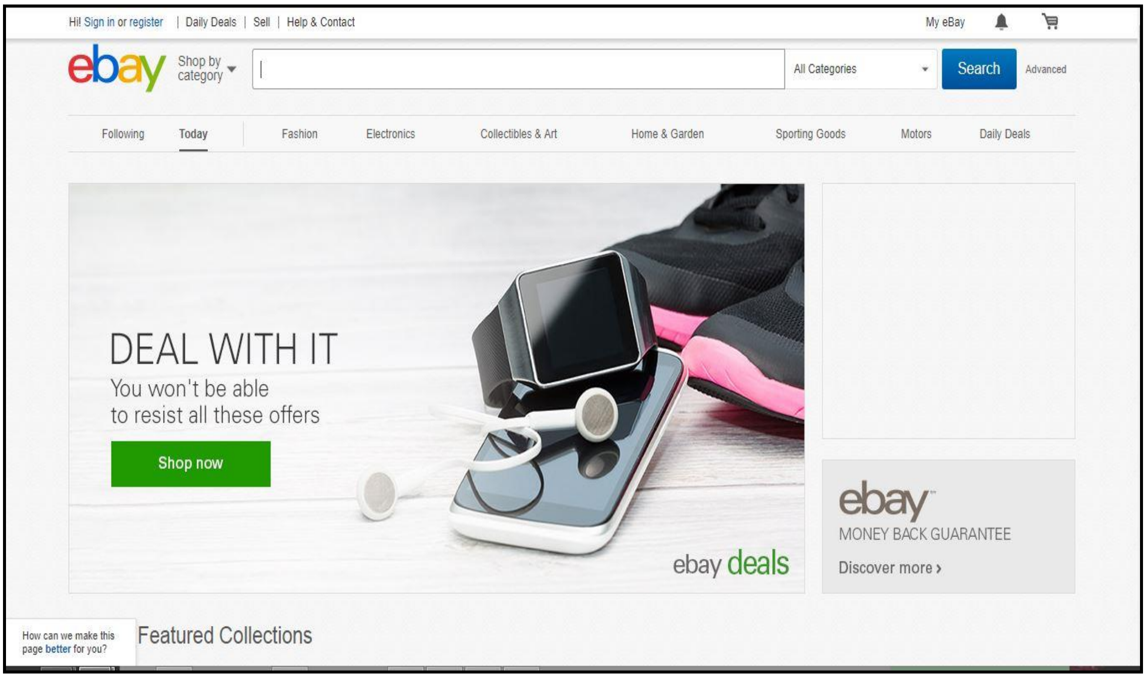


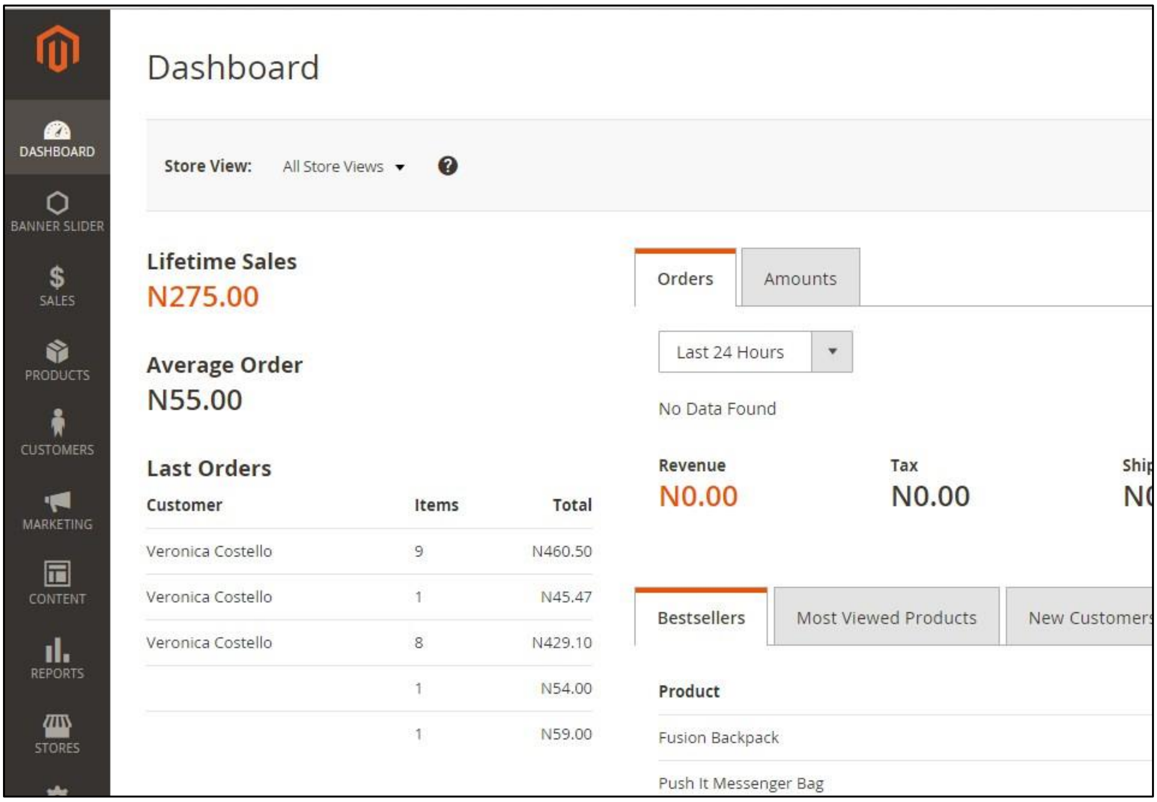
Figure 2.2: User Interface of ebay.com

2.4.2 Magento

Magento is one of the best open source e-commerce software solution which consists of customer view (website) and company view (system). Sample screen is shown below by Figure 2.3.

Features

- Able to manage the website according to their preferences.
- Online payment facility.
- Provides online customer care support.



Dashboard

Store View: All Store Views

Lifetime Sales
N275.00

Average Order
N55.00

Last Orders

Customer	Items	Total
Veronica Costello	9	N460.50
Veronica Costello	1	N45.47
Veronica Costello	8	N429.10
	1	N54.00
	1	N59.00

Last 24 Hours

No Data Found

Revenue	Tax	Ship
N0.00	N0.00	N(

Product
Fusion Backpack
Push It Messenger Bag

Figure 2.3: User Interface of Magento admin panel

2.4.3 MyDeal.lk

MyDeal has become the best and most popular online shopping website in Sri Lanka. Sample screen is shown below by Figure 2.4.

Features

- Compare and purchase items
- Delivery facility and can generate reports according to user’s wishes.

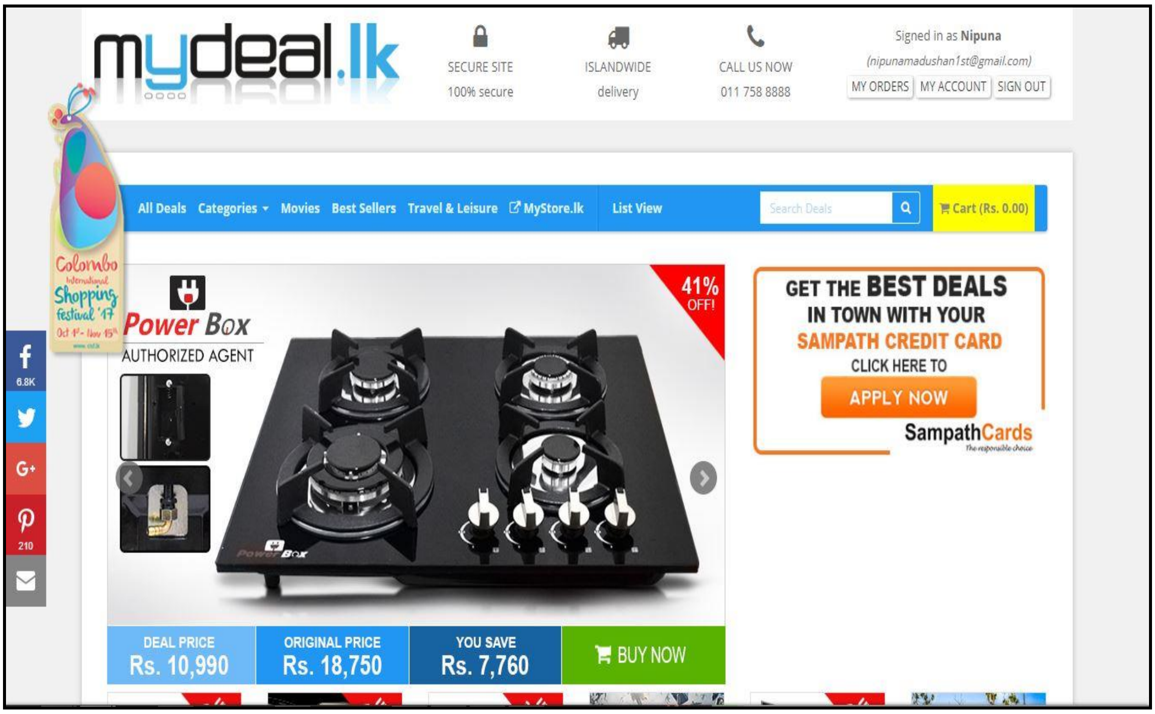


Figure 2.4: User Interface of MyDeal.lk

2.5 Functional requirements

Functional requirements of a system explains the ‘what system must do’. It simply means the functionalities and services of the system and its components.

2.5.1 Stakeholder management

Stakeholders are the people who are concerned and interested with the system such as customers, system users, suppliers, courier services etc.

- Register employees, suppliers and other system stakeholders to the system and administrator can manage the details of the system stakeholders.
- Customer registrations and they can manage their details by their own.
- Email and SMS confirmations of the stakeholder registrations.

2.5.2 Product and inventory management

Product management consists of the items and goods that sell to the customer.

- Products and Product categories can be define.
- Create and manage items with the proper descriptions of the features.
- Items can be modify according to the changes in the newly arriving items.

2.5.3 Order management

It is a process of managing the business functions which are occurred due to the orders placed for goods and services.

- Customers can place orders as well as modify orders whether it delivered or not.
- Customer can make their payments relevant to the orders online or cash on delivery.
- Can make inquiries of the orders and also can check the status of the orders of the customers.
- Customer can give feedback and ratings relevant to the service.
- Customer can add items to a cart and checkout.
- Customer can view their order history.

2.5.4 Report center

- System allows administrator to get reports according to criteria such as date range, status or customer wise reports can be taken.
- Financial reports like income, expenditure, profit and loss can be generated through the system.
- Following reports also can be generated.
  - Order details.
  - Registered customer details.
  - Best selling items.
  - Monthly transaction details.
  - Income and expenditure reports.
  - Profit and loss reports.

2.5.5 Customer Inquiry

- Customer can chat with the Ecart.lk admin / officer to make their inquiries about the services and orders.

2.6 Non - functional requirements

Non-functional requirements of a system simply means ‘what are the properties of the system’ which explain the time constraints, security and safe guard constraints.

- **Performance**

  Performance is mainly concern with the response time of the system after a transaction or any user task which is perform within the system.

- **Security and safe guard**

  System consists of user personal data, transaction details and other financial details. Therefore the security and the safe guard of the system is really important.

- **Usability**

  Usability can be concern with the ease of use and easy to learn the system.

- **Availability**

  System should be available for the use whenever the user wants to access the system.

- **Platform independence**

  System should be work on any operating system while it can be access on the any of web browser.

- **Accountability**

  System should be responsible for the transactions and what we are done in the system.

2.7 Selected process model

Process model is the development approach which split the whole system developments process in to specific stages. As the requirement from the client are stable waterfall model is used as the process model.

- **Requirement gathering**

  Requirements and business process information are collected using requirement gathering techniques. Requirement can be collected mostly by discussing with the employees and by examine the existing system.

- **Requirement analysis**

  Collected requirements are analyzed using various techniques. In this stage, determine the user expectation for the new product.

- **System design**

  Analyzed requirements and user expectations are converted into models and make conceptual design for overall system.

- **Implementation**

  In this stage conceptual designs are converted in to the actual product using programming languages.

- **Testing**

  Testing methods like user acceptance testing, beta testing, integration testing and penetration testing can be done here to verify that the developed system is fulfilling the customer requirement.

- **Maintenance**

  Implementing the system in the client site and maintaining system is done here. And also ad hoc change request from the client are address in this stage

Chapter 03 – Design

3.1 Introduction

System~~s~~ design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. There are multiple approaches for system designing. However the most widely used methods are the Object Oriented Design methods.

3.2 Alternative solutions evaluation

Standalone system may be inconvenient for users as the system cannot be accessed at any place as it has to be installed to the computer before it is being used. Clients mainly requested for a web based system as it will be very convenient for them as they can use the system at any place even when they are at home or office or out of region.

Most of the functionalities of the system may be achievable by licensed software but those web based software are not freely available and have to purchase them. At the same time these software are not adaptable to use as a software for an association; therefore key features of the system may be lost.

Before taking the decision of implementing the web based software solution, few alternative solutions suggestions were considered.

- Customize open source software solutions such as Magento and PrestaShop suite to the client requirements.
- Purchasing a commercial online shopping system and change the company business processes according the purchased system.
- Maintain orders and inventories in a separate standalone applications while advertising the goods and items to be sold in the social media

3.3 Selected solution description and justification

The client is more concern about carrying out their own business processes rather than adopting to a new system. Therefore, first and second alternatives are discarded as they do not fulfill client’s requirement. Client interest is with a web based software solution, not a standalone applications.

In order to fulfill the client’s expectations, proposed web based software solution was implemented from the scratch. Data and information of customers, orders and inventories are stored in a central database where whole business functionalities can be manage from a one system.

3.4 Object orient approach

One of the main principles in the object oriented (OO) approach is that of abstraction, not of data structures and processes separately but both together. An object is a set of data structures and the methods or operations needed to access those structures. [2]

Object Oriented Designing (OOD) is a method that uses objects to develop a system. Each object interacts with each other and they have their own states and operations. Unified Modeling Language is one of the most popular methods used in order to develop systems with OOD concept. There are some object models defined by UML such as Use Case Diagrams, Class Diagrams, and Sequence Diagrams that aid system designing process.

3.5 Object orient diagrams

3.5.1 Use Case Diagram

Use case diagrams symbolize the user interaction with the system while addressing the functional requirements of the system. It is a representation of a user’s interaction with the system that shows the relationship between the users. It contains use cases, actors, their relationship, and system boundary. In the Ecart.lk Online Shopping System there are main 3 actors of the system named administrator, officer and online customer. The use case shows function of the system, actor is a person or system that interact with the central system. There are different type of relationship among use cases and actors such as association, generalization, include and extend. The system boundary represents scope of the system by rectangle around the use cases. The Figure 3.1 shows use case diagrams of the system.

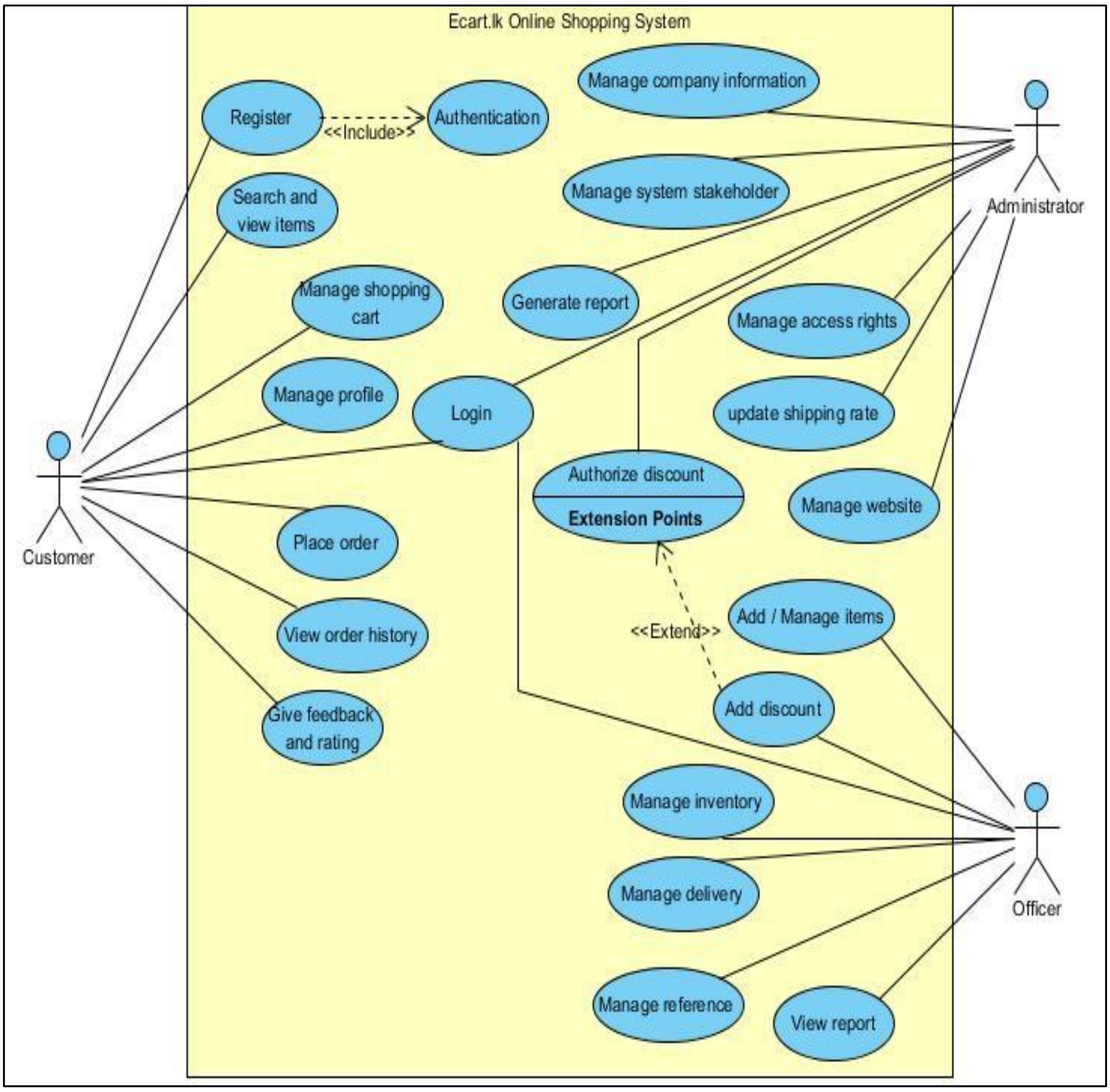


Figure 3.1: Use Case Diagram for the System

3.5.2 Class Diagram

Class diagram is one of the most important diagram in the UML. It shows the detailed database structure / classes of the system. Class diagram represent classes interfaces and relationship among those entities. Classes are represent with a box which is divided into 2 parts. First part shows attributes and second part is methods. Following Figure 3.2 shows class diagram of the system.

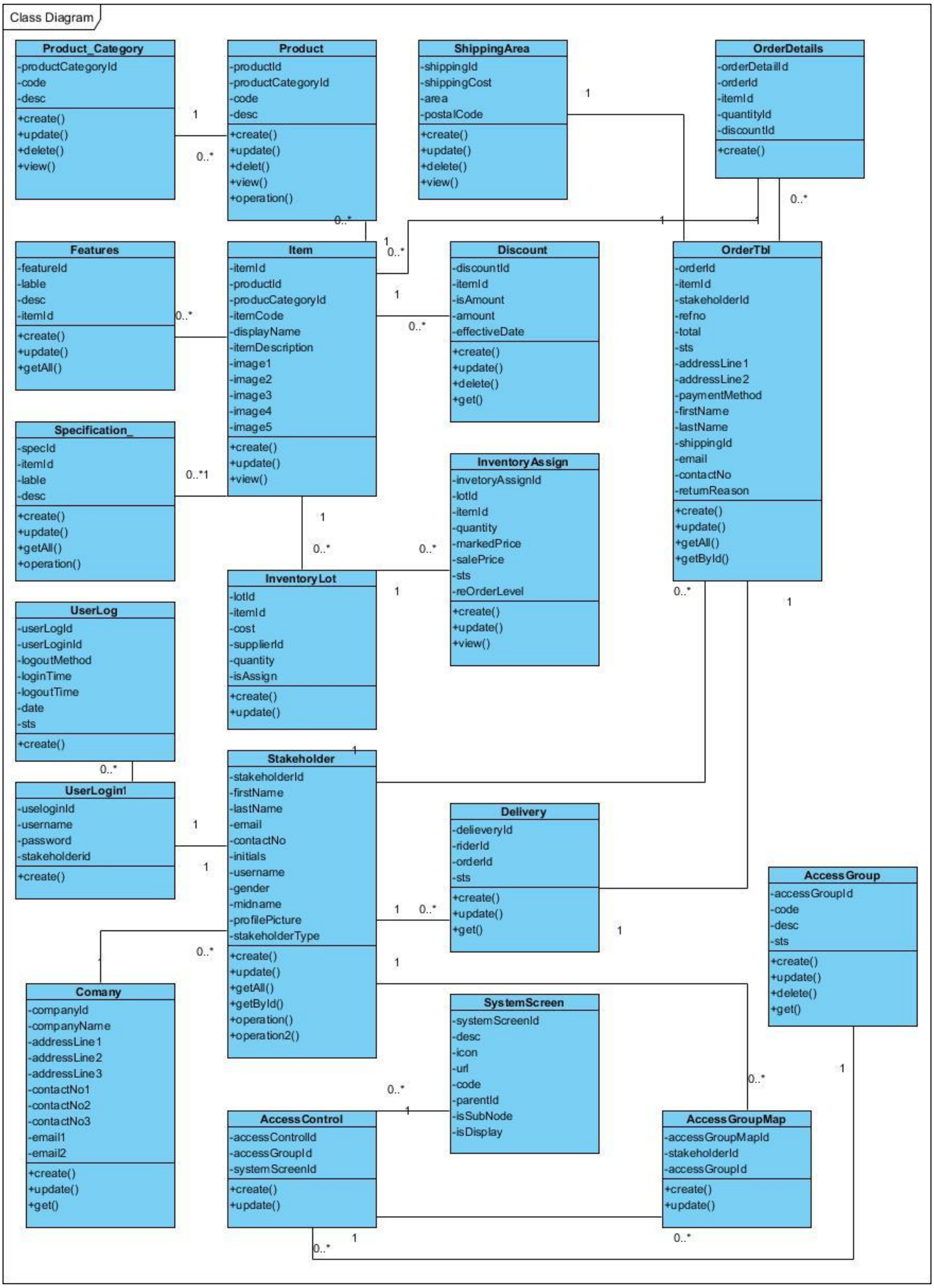


Figure 3.2: Class diagram for the System

3.5.3 Activity Diagram

Activity diagram is UML diagram which represent the flow of control of overall system or part of a system. This flow can be sequential or parallel. Activity diagrams are used for business process modeling. This OO diagram describe the dynamic aspect of the system. There are some defined notations for activity diagrams such as rounded rectangle represent actions, diamonds represent decision, black cycle represent the start of the activity, an encircled represent the end of the activity, swim lane represent the portioning the actor in the activity diagrams. Figure 3.3 shows activity diagram of the system.

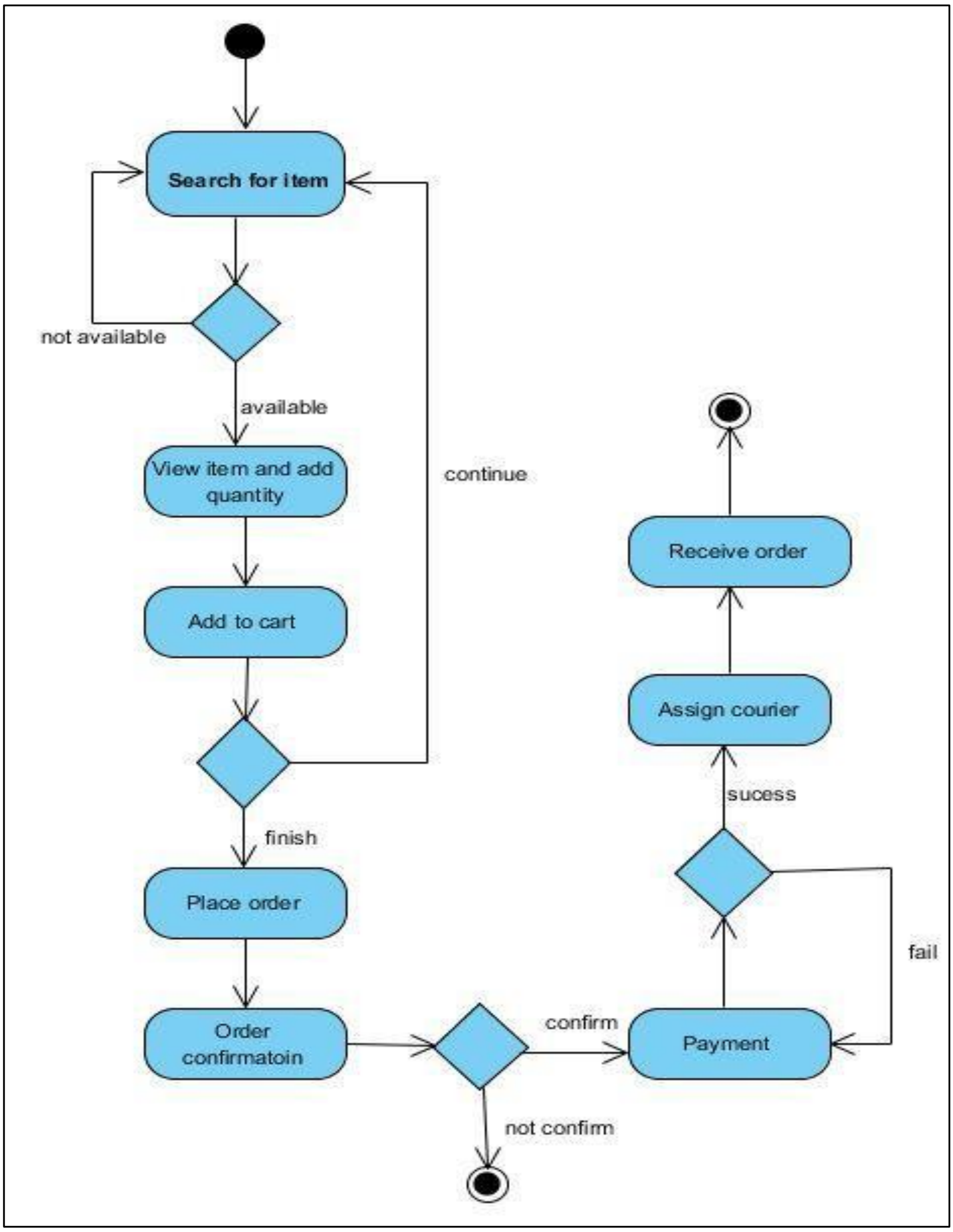


Figure 3.3: Activity diagram of the system

3.5.4 Sequence Diagram

It is an interaction diagram which shows how separate objects operate with each other in what order is called sequence diagram. In another word it can be called as flow of object in order in the system. It shows the object arrangement in a time sequence. Following Figure 3.4 show the sequence diagram of the developed system between the objects customer, system, cart and order.

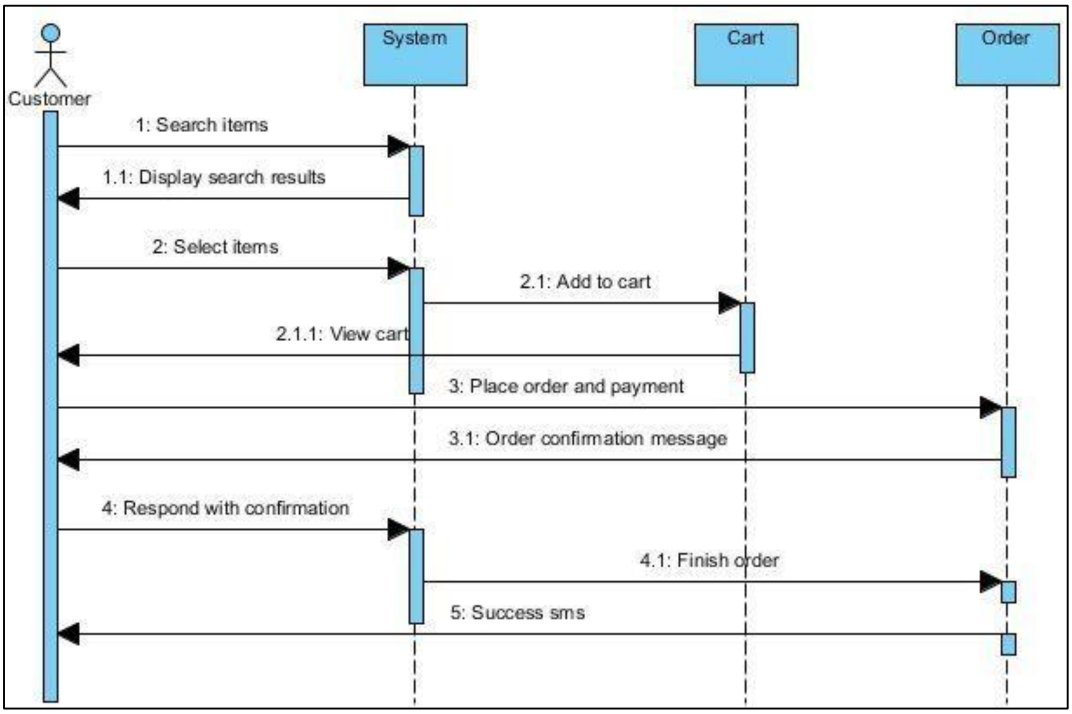


Figure 3.4: Sequence diagram of the system

3.5.5 ER Diagram

Entity Relationship diagram is a data model for describing the relationships of entity set stored in the database. There are three key elements of an ERD. There are entities, attributes and relationship. Entity is a thing normally called as table. Attributes provide details about concept in each entity. Relationship shows how entities relate to each other. Associate entities are optional and it needed when there is a many-to-many relationship among entities. Figure 3.5 shows ER diagram of the data design.

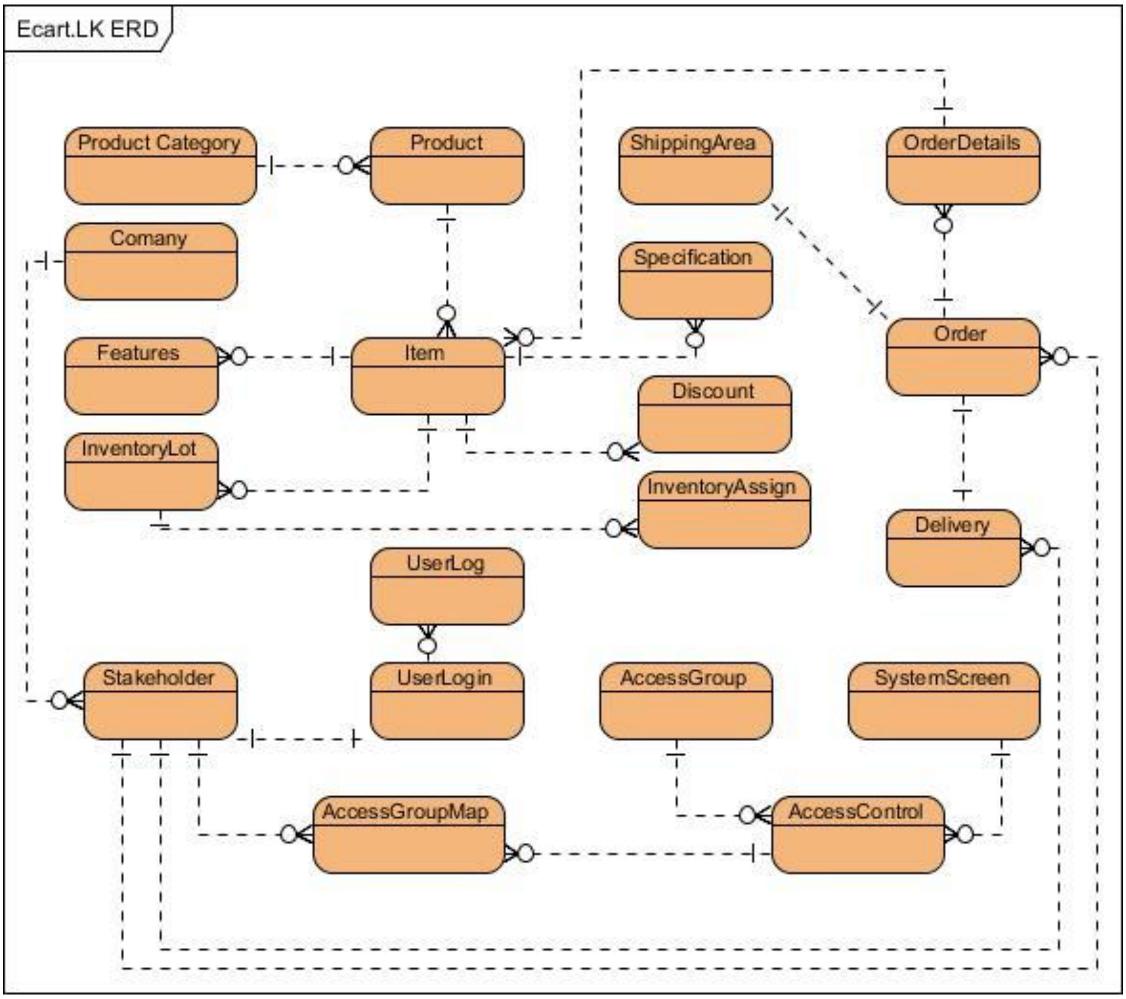


Figure 3.5: ER Diagram for the System

3.6 Table structure

The following Table 3.1, Table 3.2, Table 3.3 shows the table structure of the product category, product and the item tables. All the other tables using in the system are same as follow structure.

E – Entity R – Reference D - Derive

Attribute	Description	Data Type	Length	Key	Type
PRDCATID	Product category Id	INTEGER	11	PK	E
DESCR	Description	VARCHAR	255		E
CODE	Code	VARCHAR	255		E

CREATED_USER_ID	Created user Id	INTEGER	11	FK	R
CREATED_DATE	Created date	DATE			E
LUP_DATE	Last updated date	DATE			E

Table 3.1: Table structure for product category table

Attribute	**Description**	**Data Type**	**Length**	**Key**	**Type**
PRODUCTID	Product Id	INTEGER	11	PK	E
PRDCATID	Product category Id	INTEGER	11	FK	R
CODE	Code	VARCHAR	255		E
NAME	Name	VARCHAR	255		E
CREATED_USER_ID	Created user Id	INTEGER	11	FK	R
CREATED_DATE	Created date	DATE			E
LUP_DATE	Last updated date	DATE			E

Table 3.2: Table structure for product table

Attribute	**Description**	**Data Type**	**Length**	**Key**	**Type**
ITEMID	Item Id	INTEGER	11	PK	E
PRDCATID	Product category Id	INTEGER	11	FK	R
PRODUCTID	Product Id	INTEGER	11	FK	R
DISPLAYNAME	Display Name	VARCHAR	255		E

ITEMDESCRIPTION	Item Description	VARCHAR	255		E
ITEMCODE	Item Code	VARCHAR	255		E
IMAGE1	Image 1	LONG BLOB			E
IMAGE2	Image 2	LONG BLOB			E
IMAGE3	Image 3	LONG BLOB			E
IMAGE4	Image 4	LONG BLOB			E
IMAGE5	Image 5	LONG BLOB			E
CREATED_USER_ID	Created user Id	INTEGER	11	FK	R
CREATED_DATE	Created date	DATE			E
LUP_DATE	Last updated date	DATE			E

Table 3.3: Table structure for Item table

3.7 User Interface Design

User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface. Today, user interface is found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc. User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction. UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both. [3]

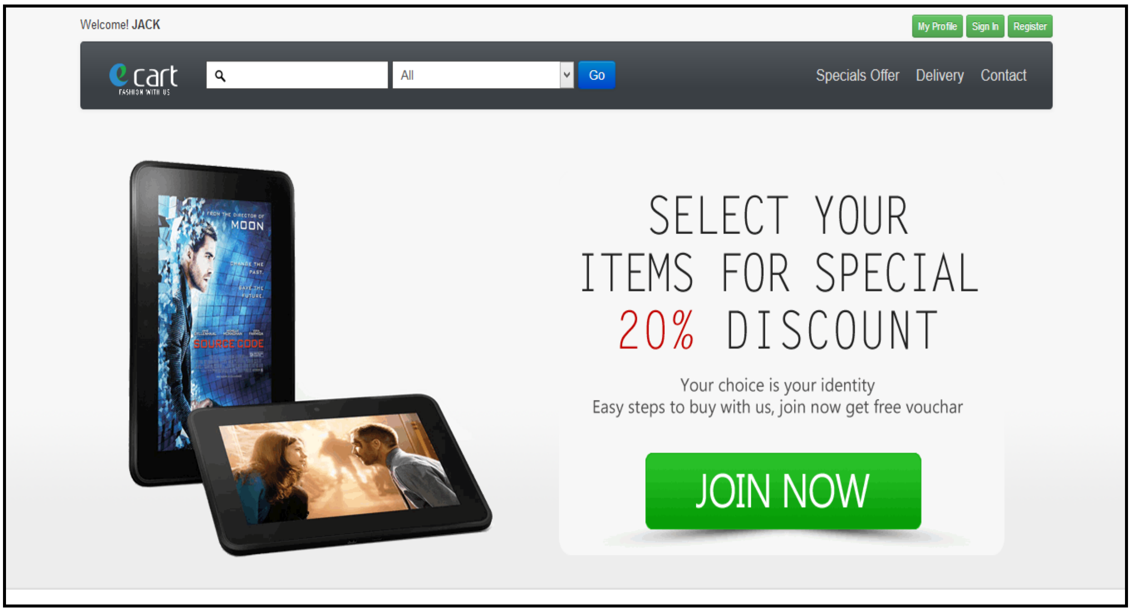
User interface should be,

- Ease of use
- Attractive
- Ease of understanding

- Quick response time
- Easy to navigate

Home Page – Website

Online shopping system mainly consist of 2 modules which is admin system and the website. Following Figure 3.6 shows the home page of the website which can be access locally by the URL https://localhost:8080/ecartlk. But after it is hosted it can access by the URL https://ecart.lk. Home page and the whole website is developed with very simple UI and professional look and feel. User can easily navigate through the website so that he can get the maximum use of the online shopping facility. Items can be classified under several categories and products (sub categories). Categories such as electronic items, beauty and health, accessories, automobile and products like table fans, LED TV, motor bike, van are provide in the UI to improve the user friendliness in the system. That is shown in the following Figure 3.7.



SELECT YOUR ITEMS FOR SPECIAL 20% DISCOUNT

Your choice is your identity
Easy steps to buy with us, join now get free vouchar

Figure 3.6: Home Page – Website

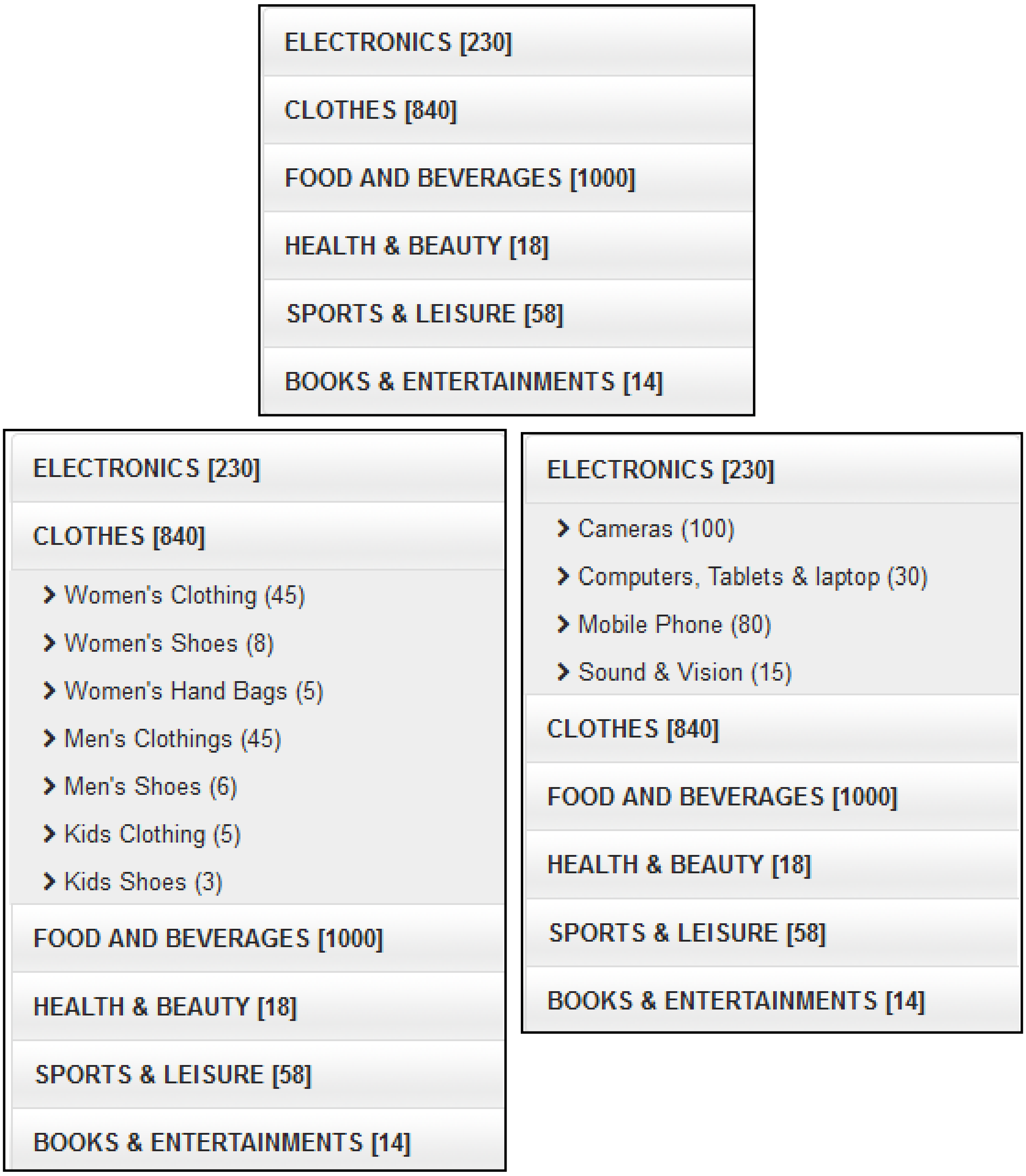


Figure 3.7: Product Categories and Product

Home Page – System (Admin panel)

System home page can be accessed by the URL https://localhost:8080/ecartlk/system only after login in to the admin panel. From the above URL it direct to the login page and if the credentials are correct then the system (admin panel) is run in a separate popup window. That can be find in the Figure 3.8.

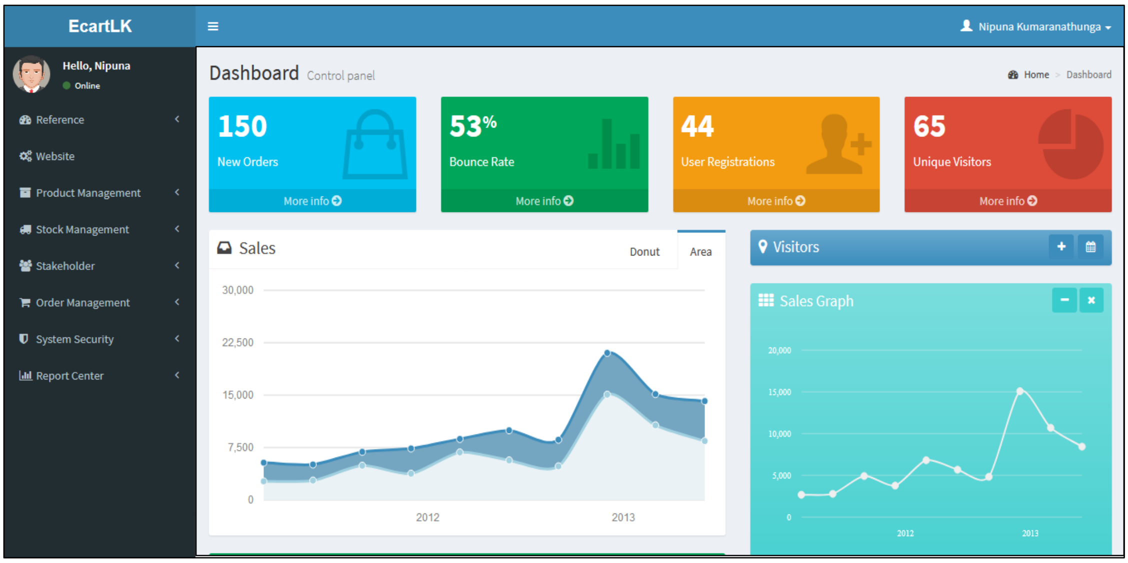


Figure 3.8: Home Page system

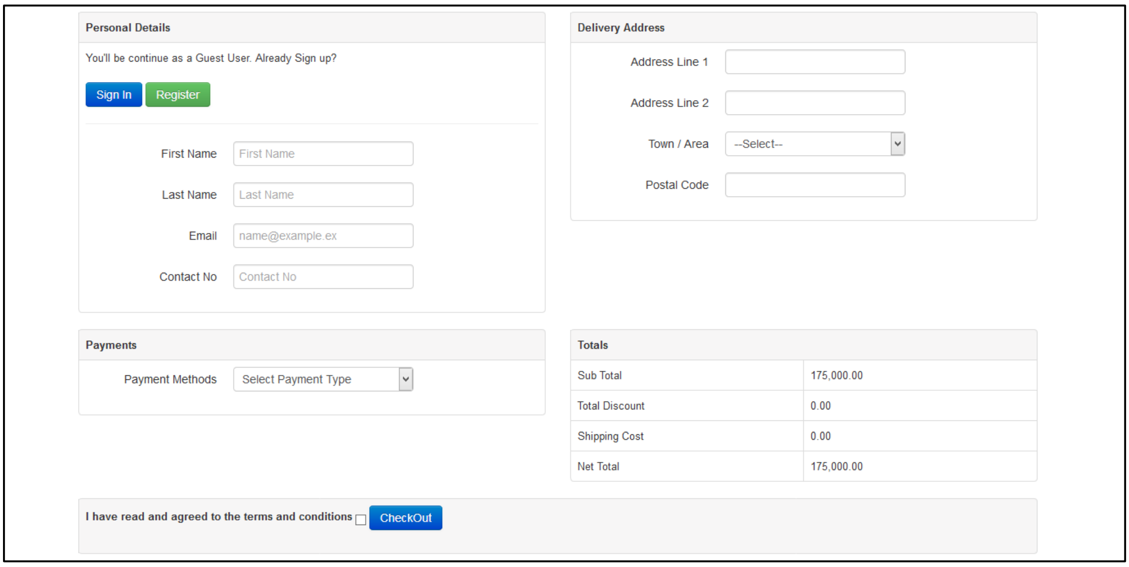
3.8 User interface standard

Front end development or UI development was done by the HTML and CSS used as the style sheet language. Bootstrap is used as a front end framework which has built with HTML and CSS. Client side scripting language is java script that is implement in this project.

Input Forms

Mainly input forms contains input fields and tool bar with buttons. Data entered to input fields by text fields, drop downs, check box etc., Fields and buttons are disable or enable with respect to the CRUD operations. Following are some example for input data forms.

1. Below Figure 3.9 shows the checkout screen which entered information like personal details, shipping details and payment methods.



Personal Details

You'll be continue as a Guest User. Already Sign up?

Sign In | Register

First Name: First Name

Last Name: Last Name

Email: name@example.ex

Contact No: Contact No

Delivery Address

Address Line 1:

Address Line 2:

Town / Area: --Select--

Postal Code:

Payments

Payment Methods: Select Payment Type

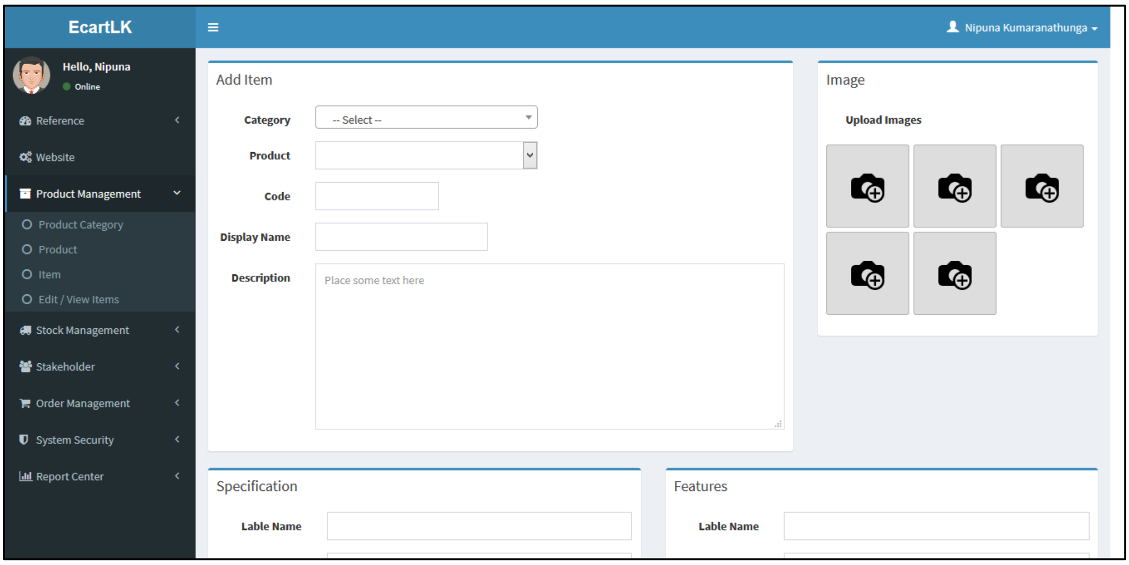
Totals

Sub Total	175,000.00
Total Discount	0.00
Shipping Cost	0.00
Net Total	175,000.00

I have read and agreed to the terms and conditions □ CheckOut

Figure 3.9: Order Checkout screen

2. Following Figure 3.10 shows the item add screen which provide the facility to add images to the adding item and also other input fields for general details.



Add Item

Category: -- Select --

Product:

Code:

Display Name:

Description: Place some text here

Image

Upload Images

Specification

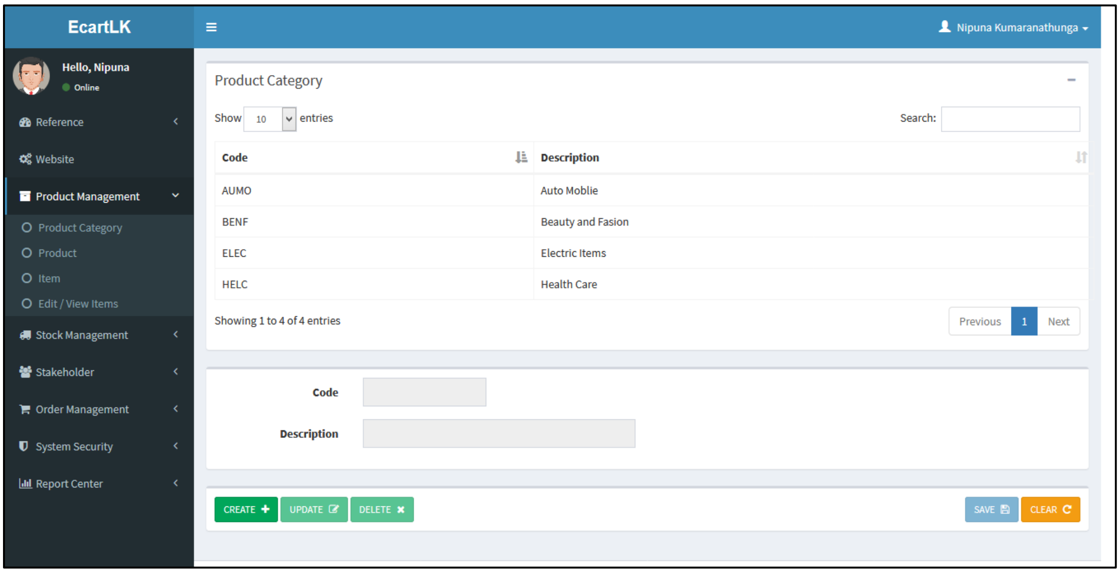
Lable Name:

Features

Lable Name:

Figure 3.10: Item Add screen

3. Figure 3.11 is a simple reference screen with button tool bar.



Code	Description
AUMO	Auto Moblie
BENF	Beauty and Fasion
ELEC	Electric Items
HELC	Health Care

Showing 1 to 4 of 4 entries

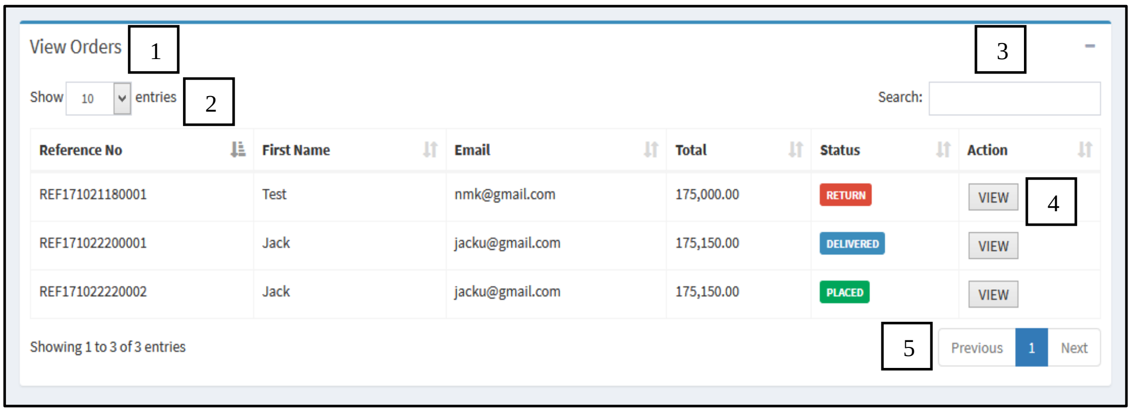
Code

Description

Figure 3.11: Product Category screen

Data Tables

Data tables / grids use to show the saved data in the database and also use to store data temporary before submit into the database. Datatable.js JavaScript library is used to create tables which has several features. Figure 3.12 shows the component of the data table.



Reference No	First Name	Email	Total	Status	Action
REF171021180001	Test	nmk@gmail.com	175,000.00	RETURN	VIEW
REF171022200001	Jack	jacku@gmail.com	175,150.00	DELIVERED	VIEW
REF171022220002	Jack	jacku@gmail.com	175,150.00	PLACED	VIEW

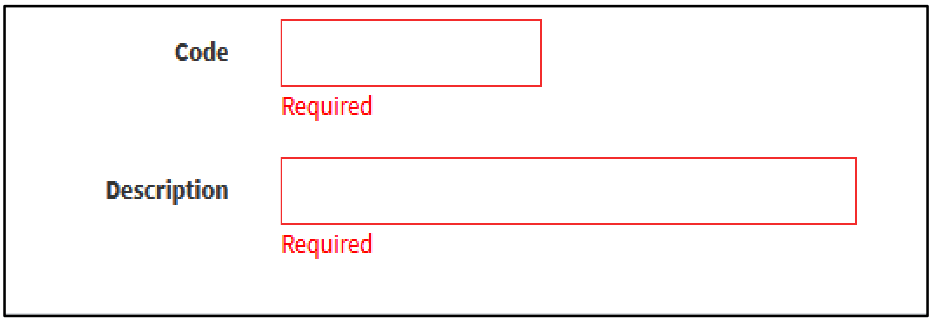
Showing 1 to 3 of 3 entries

Figure 3.12: Data table

1. Table name.
2. Number of rows shows at once in the table.
3. This can search any key word in any column in the data table.
4. Buttons that execute operation row wise.
5. Table paging which partition data in to pages.

Validation

Validation of a system is an important factor and Ecart.lk Online Shopping System is validated 100% to ensure the authorization and authentication. In terms of input field validation, following Figure 3.13 shows an example. Error is indicate just below the relevant field and the field is also getting red color. Any of data forms cannot be submit if those errors message are shows in the screen.



Code

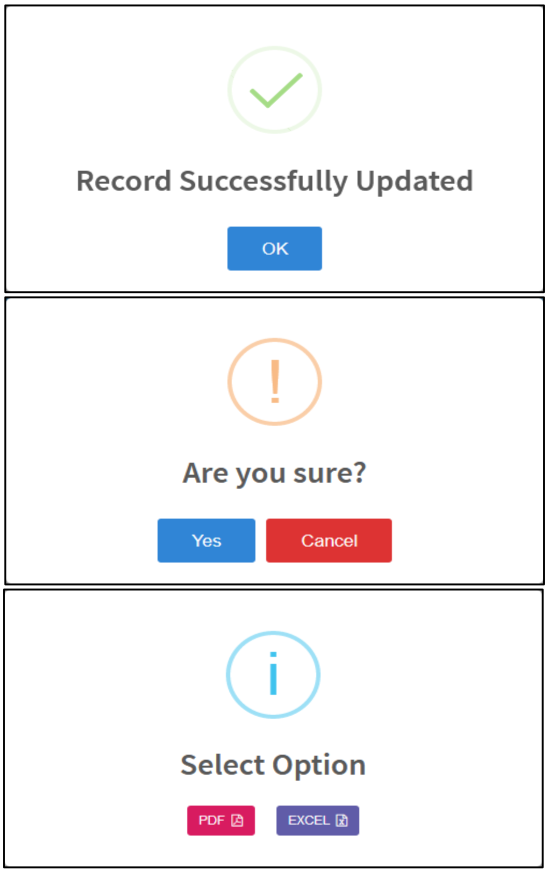
Required

Description

Required

Figure 3.13: Validation

Messages, dialog box and popup message



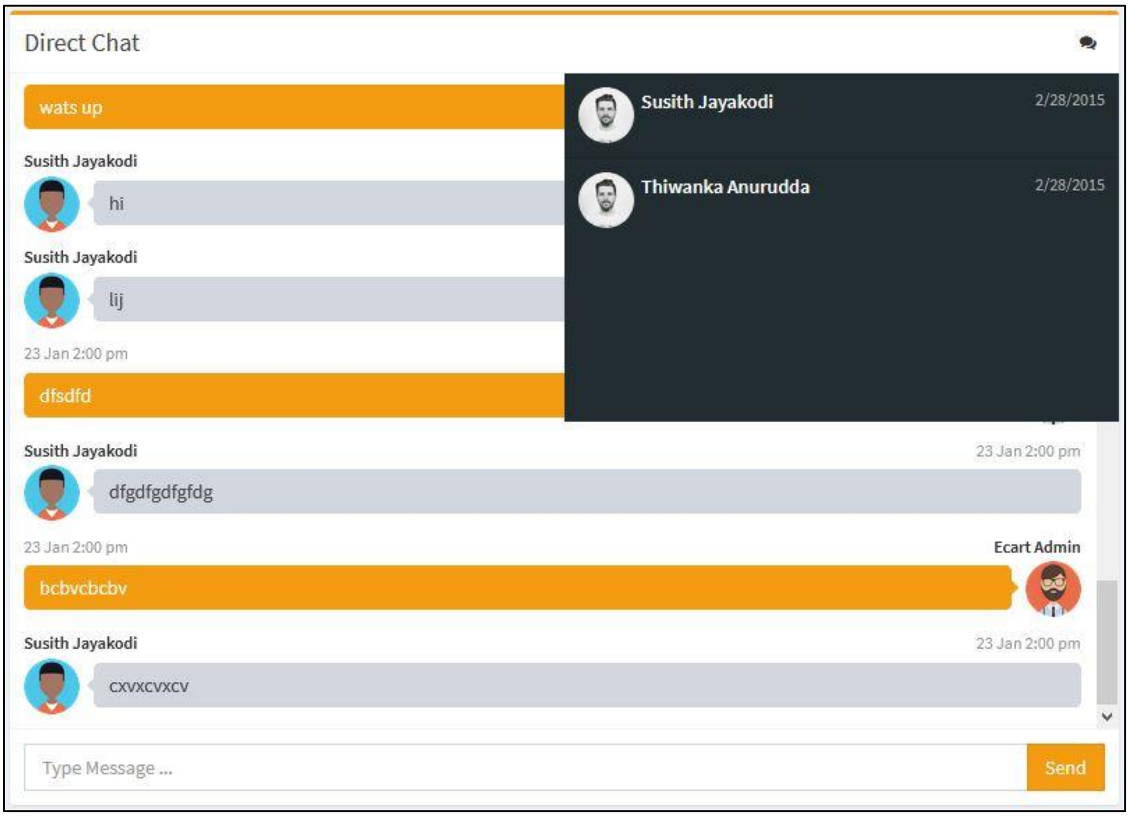
Record Successfully Updated

Are you sure?

Select Option

Figure 3.14: Message box, alerts and popup messages

Inquiry chat component



Direct Chat

wats up

Susith Jayakodi

hi

Susith Jayakodi

lij

23 Jan 2:00 pm

dfsdfd

Susith Jayakodi 23 Jan 2:00 pm

dfgdfgdfgfdg

23 Jan 2:00 pm Ecart Admin

bcbvcbcbv

Susith Jayakodi 23 Jan 2:00 pm

cxvxcvxcv

Susith Jayakodi 2/28/2015

Thiwanka Anurudda 2/28/2015

Figure 3.15: Chat screen admin panel

Chapter 04 – Implementation

Implementation is the longest and most time consuming phase in software development life cycle. Successful design phase prior to implementation phase always make the way to success the development phase. All of the works are divided into models. Each model transfer into system by using code, this is the main task of the programmers.

This chapter is mainly focus on the implementation details of the developed system. In addition to that, this section will describe the hardware/software environment, development tools as well as model/code structure and reusable code.

4.1 Development environment

Java Enterprise Edition (JavaEE / J2EE) is selected as the major programming language to implement this Ecart.lk Online Shopping System as it is a web based project. MVC design pattern is used to develop this project. This whole system is rum on a multi-tier architecture and following Figure 4.1 shows the overview diagram.

HTML, JSP, JQuery, JS and Ajax are used as the presentation layer and Struts used to validate the presentation layer as the presentation logic layer. Duty of spring framework is dependency injection which is business layer. ORM integrate in the system is hibernate and that is data access layer. MySQL used as the database management system for the Online Shopping System.

Apache Struts is a free, open-source, MVC framework for creating elegant, modern Java web applications. It favors convention over configuration, is extensible using a plugin architecture, and ships with plugins to support REST, AJAX and JSON. [4]

Hibernate is ORM framework which gives several features when deals with the database.

1. Data roll back.
2. Do not need to handle exception
3. All tables can be created automatically by using the Hibernate domain classes.

4.2 Hardware and software requirements

Selecting correct and compatible versions of software and hardware is really important when developing a system.

Hardware requirements

- Personal computer or a laptop Dual Core CPU or above.
- Memory: minimum 2GB RAM or above.
- Internal hard disk: 100GB or above.
- Internet connection either modem or dongle.
- Keyboard and mouse.

Software Requirement

- Application server: Apache tomcat server 7.0.23
- Java: JDK 7
- Code editing IDE: Spring Tool Suit
- Struts 1.2
- Spring framework 3.2
- Hibernate ORM 3.6
- Database: MySQL server 5.6
- Database browser: Navicat Premium
- Graphical designing: Adobe Photoshop
- Report : Java jxl API for excel and iText library for PDF creation
- Web browser: Chrome or Firefox latest versions are recommended

4.3 Client / server environment

Same as selecting the development system requirements, selecting the client and server environment is important. Following tables at Table 4.1 and Table 4.2 are the minimum hardware and software requirements needed.

Server-side Environment

Hardware Requirement	Software Requirement
50GB hard disk space	Apache Tomcat 7.0.23
4GB RAM	JDK 7
Personal computer with Dual Core CPU	64 bit Operating System

Table 4.1: Server side minimum requirements

Client-side Environment

Hardware Requirement	**Software Requirement**
20GB hard disk space	PDF reader
2GB RAM	Web browser (ex: Chrome, Firefox)
Printer	Excel viewer

Table 4.2: Client side minimum requirement

4.4 Development tools

Development tools and IDE are used in the process of implementation of the system to accelerate processes. Other than developing tools, additional third party tools were used for several purposes.

4.4.1 Spring Tool Suite (STS)

The Spring Tool Suite is an Eclipse-based development environment that is customized for developing Spring applications. It provides a ready-to-use environment to implement, debug, run, and deploy your Spring applications. [5]

STS is support for the java and spring applications, automated code generations, code formatting and JUnit testing can be done. Spring Tool Suite version 3.5.1 used for the development.

4.4.2 Navicat Premium

Navicat premium is database development and management tool that facilitate any kind of SQL server such as MySQL, Oracle SQL, PostgreSQL etc., this supports several database functions.

- Data migration facilities.
- Diversified data manipulation tools.
- Easy SQL building and editing facility.
- Create and manage databases.
- Backup management.

- Advanced secure connection through SSH tunnel.

4.4.3 Visual Paradigm

Visual Paradigm for UML 8.0 Enterprise Edition is designed for UML diagram modelling. This facilitate to draw all UML diagrams like use case diagrams, data flow diagrams, activity diagrams, ER diagrams and sequence diagrams.

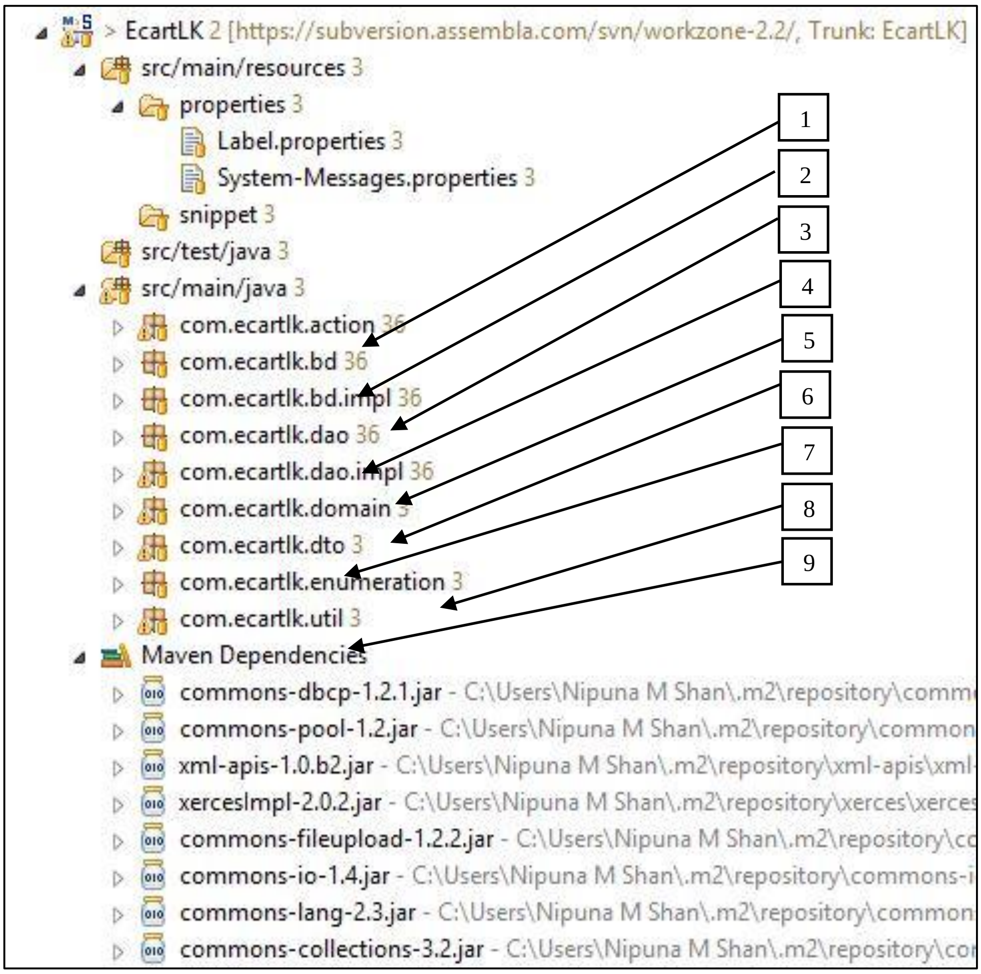
4.4.4 Microsoft Project 2013

MS project was used to create WBS (Work Breakdown Structure) of the system.

4.5 Coding standards and structure

Project structure of the developed system can be categorized as folder structure and package structure. Given below sections provide the details about these structures.

4.5.1 Package structure



```
> EcartLK 2 [https://subversion.assembla.com/svn/workzone-2.2/, Trunk: EcartLK]
  src/main/resources 3
    properties 3
      Label.properties 3
      System-Messages.properties 3
    snippet 3
  src/test/java 3
  src/main/java 3
    com.ecartlk.action 36
    com.ecartlk.bd 36
    com.ecartlk.bd.impl 36
    com.ecartlk.dao 36
    com.ecartlk.dao.impl 36
    com.ecartlk.domain 3
    com.ecartlk.dto 3
    com.ecartlk.enumeration 3
    com.ecartlk.util 3
  Maven Dependencies
    commons-dbcp-1.2.1.jar - C:\Users\Nipuna M Shan\.m2\repository\comm
    commons-pool-1.2.jar - C:\Users\Nipuna M Shan\.m2\repository\common
    xml-apis-1.0.b2.jar - C:\Users\Nipuna M Shan\.m2\repository\xml-apis\xml
    xercesImpl-2.0.2.jar - C:\Users\Nipuna M Shan\.m2\repository\xerces\xerces
    commons-fileupload-1.2.2.jar - C:\Users\Nipuna M Shan\.m2\repository\cc
    commons-io-1.4.jar - C:\Users\Nipuna M Shan\.m2\repository\commons-i
    commons-lang-2.3.jar - C:\Users\Nipuna M Shan\.m2\repository\common
    commons-collections-3.2.jar - C:\Users\Nipuna M Shan\.m2\repository\co
```

Figure 4.1: Package structure of the source code

Figure 4.1 shows the package structure of the source code and description about the numbered areas are given below.

1. **Action classes**

   Action class transfer data from the front end to specific business processes after manipulate. Its duty is to transfer between presentation layer and business layer.

2. **Business Delegate classes**

   This package contain the interface classes which is act as business delegate layer. It use to decouple presentation layer and business layer.

3. **Business Delegate implementation classes**

   Contain the implementation methods of the relevant interface classes in the business delegate layer.

4. **Data Access Object classes**

   This package contains Data Access Object interface classes and defined abstract methods.

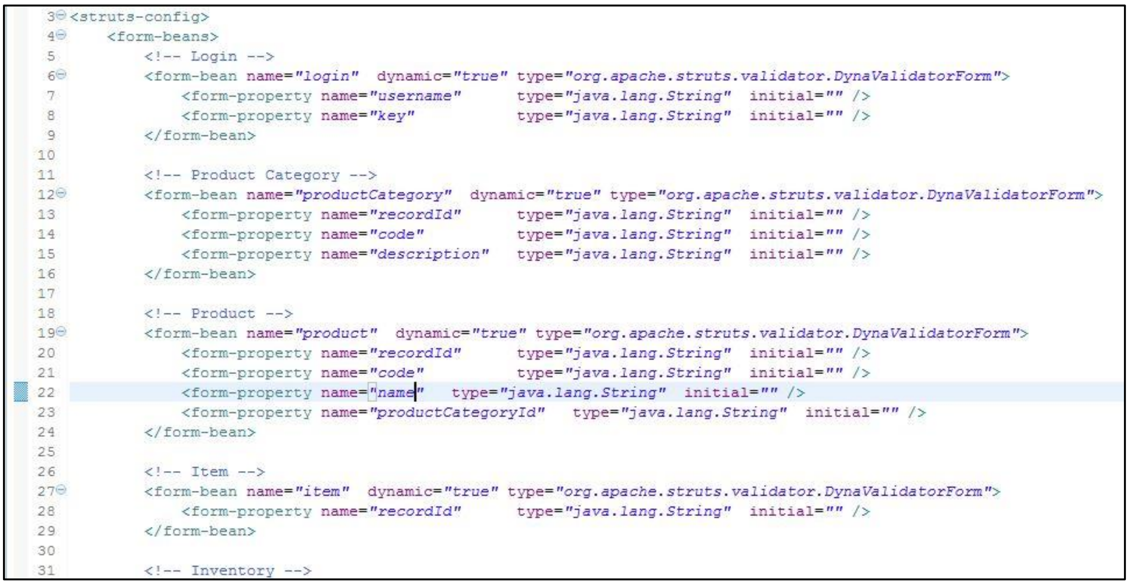
5. **Data Access Object implementation classes**
   Contains the implementation methods of the relevant interface classes in the business delegate layer.
6. **Domain classes**
   Contain the classes that represent the table in the database.
7. **Data Transfer Object classes**
8. **Enumeration classes**
9. **Utility classes**
   Classes which have predefine processes that help for the other processes.

4.6 Open source framework

Main open source frameworks that are used in the system are Apache Struts, Spring and Hibernate. These framework technologies provide several advantages for a web base software solution.

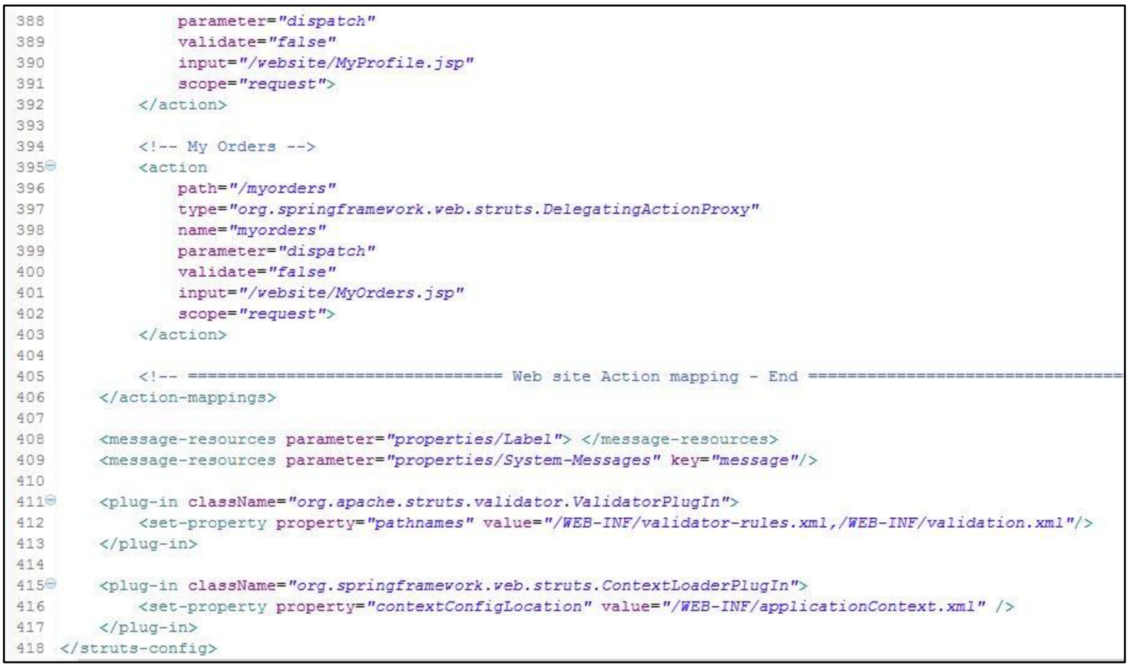
4.6.1 Apache Struts

Goal of the struts is to separate model from the view and controller. Struts provide the ActionServlet (controller) and it facilitate the writing of template for the front end or view (presentation layer). The model code is maintain the configuration file call struts-config.xml



```
<struts-config>
    <form-beans>
        <!-- Login -->
        <form-bean name="login"  dynamic="true" type="org.apache.struts.validator.DynaValidatorForm">
            <form-property name="username"      type="java.lang.String"  initial="" />
            <form-property name="key"           type="java.lang.String"  initial="" />
        </form-bean>

        <!-- Product Category -->
        <form-bean name="productCategory"  dynamic="true" type="org.apache.struts.validator.DynaValidatorForm">
            <form-property name="recordId"      type="java.lang.String"  initial="" />
            <form-property name="code"          type="java.lang.String"  initial="" />
            <form-property name="description"   type="java.lang.String"  initial="" />
        </form-bean>

        <!-- Product -->
        <form-bean name="product"  dynamic="true" type="org.apache.struts.validator.DynaValidatorForm">
            <form-property name="recordId"      type="java.lang.String"  initial="" />
            <form-property name="code"          type="java.lang.String"  initial="" />
            <form-property name="name"   type="java.lang.String"  initial="" />
            <form-property name="productCategoryId"   type="java.lang.String"  initial="" />
        </form-bean>

        <!-- Item -->
        <form-bean name="item"  dynamic="true" type="org.apache.struts.validator.DynaValidatorForm">
            <form-property name="recordId"      type="java.lang.String"  initial="" />
        </form-bean>

        <!-- Inventory -->
```

Figure 4.2: struts-config.xml - bean mapping

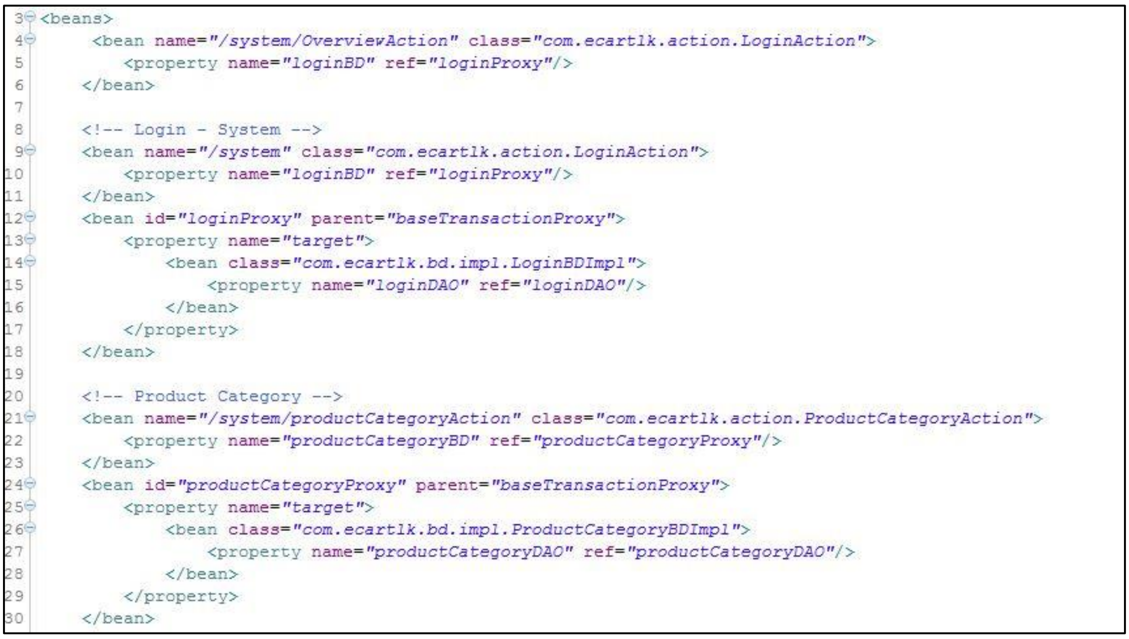


```
                parameter="dispatch"
                validate="false"
                input="/website/MyProfile.jsp"
                scope="request">
            </action>

            <!-- My Orders -->
            <action
                path="/myorders"
                type="org.springframework.web.struts.DelegatingActionProxy"
                name="myorders"
                parameter="dispatch"
                validate="false"
                input="/website/MyOrders.jsp"
                scope="request">
            </action>

            <!-- ================================ Web site Action mapping - End ================================
        </action-mappings>

        <message-resources parameter="properties/Label"> </message-resources>
        <message-resources parameter="properties/System-Messages" key="message"/>

        <plug-in className="org.apache.struts.validator.ValidatorPlugIn">
            <set-property property="pathnames" value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml"/>
        </plug-in>

        <plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">
            <set-property property="contextConfigLocation" value="/WEB-INF/applicationContext.xml" />
        </plug-in>
    </struts-config>
```

Figure 4.3: struts-config.xml - action mapping

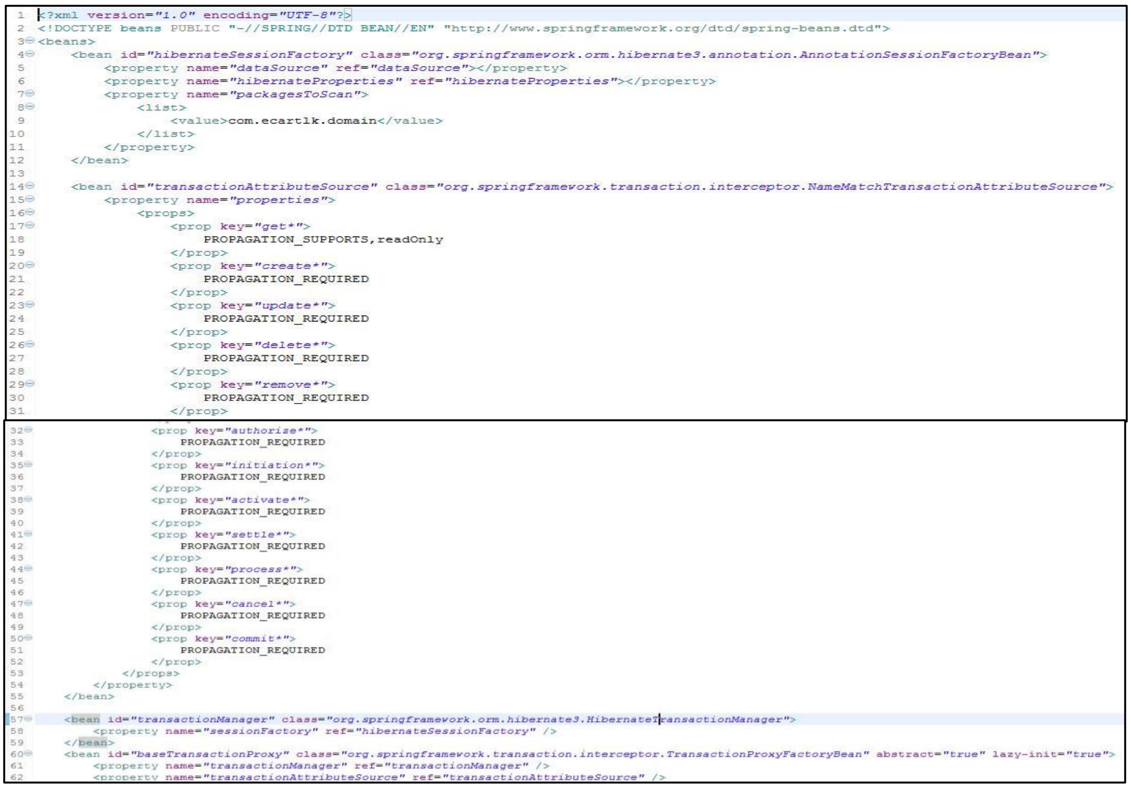
4.6.2 Spring

Spring framework was used for the dependency injection. Spring is responsible for manage the struts and hibernate. Spring configurations also wrote in xml documents. Following figures Figure 4.4 and Figure 4.5 shows the spring configuration.



```xml
<beans>
     <bean name="/system/OverviewAction" class="com.ecartlk.action.LoginAction">
        <property name="loginBD" ref="loginProxy"/>
    </bean>

    <!-- Login - System -->
    <bean name="/system" class="com.ecartlk.action.LoginAction">
        <property name="loginBD" ref="loginProxy"/>
    </bean>
    <bean id="loginProxy" parent="baseTransactionProxy">
        <property name="target">
            <bean class="com.ecartlk.bd.impl.LoginBDImpl">
                <property name="loginDAO" ref="loginDAO"/>
            </bean>
        </property>
    </bean>

    <!-- Product Category -->
    <bean name="/system/productCategoryAction" class="com.ecartlk.action.ProductCategoryAction">
        <property name="productCategoryBD" ref="productCategoryProxy"/>
    </bean>
    <bean id="productCategoryProxy" parent="baseTransactionProxy">
        <property name="target">
            <bean class="com.ecartlk.bd.impl.ProductCategoryBDImpl">
                <property name="productCategoryDAO" ref="productCategoryDAO"/>
            </bean>
        </property>
    </bean>
```

Figure 4.4: Spring configuration file



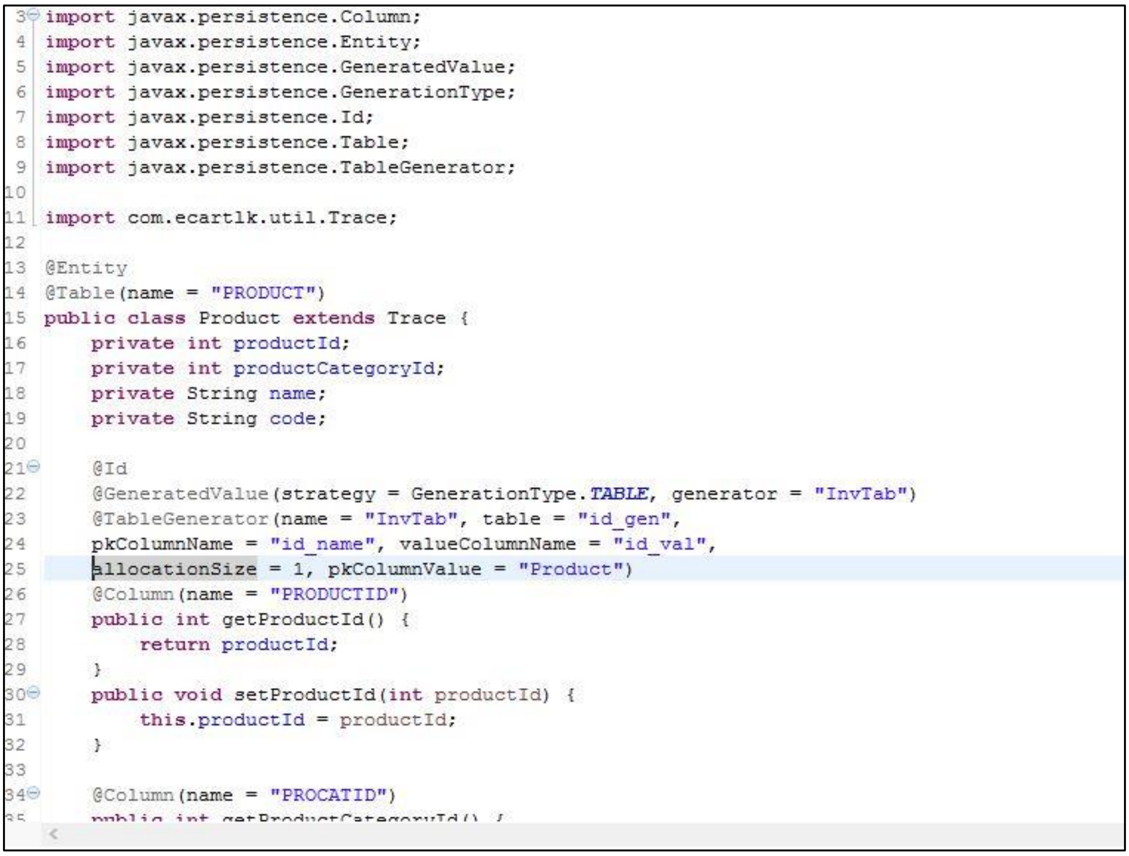
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
    <bean id="hibernateSessionFactory" class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean">
        <property name="dataSource" ref="dataSource"></property>
        <property name="hibernateProperties" ref="hibernateProperties"></property>
        <property name="packagesToScan">
            <list>
                <value>com.ecartlk.domain</value>
            </list>
        </property>
    </bean>

    <bean id="transactionAttributeSource" class="org.springframework.transaction.interceptor.NameMatchTransactionAttributeSource">
        <property name="properties">
            <props>
                <prop key="get*">
                    PROPAGATION_SUPPORTS,readOnly
                </prop>
                <prop key="create*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="update*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="delete*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="remove*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="authorize*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="initiation*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="activate*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="settle*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="process*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="cancel*">
                    PROPAGATION_REQUIRED
                </prop>
                <prop key="commit*">
                    PROPAGATION_REQUIRED
                </prop>
            </props>
        </property>
    </bean>

    <bean id="transactionManager" class="org.springframework.orm.hibernate3.HibernateTransactionManager">
        <property name="sessionFactory" ref="hibernateSessionFactory" />
    </bean>
    <bean id="baseTransactionProxy" class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean" abstract="true" lazy-init="true">
        <property name="transactionManager" ref="transactionManager" />
        <property name="transactionAttributeSource" ref="transactionAttributeSource" />
```

Figure 4.5: Spring configuration file

4.6.3 Hibernate

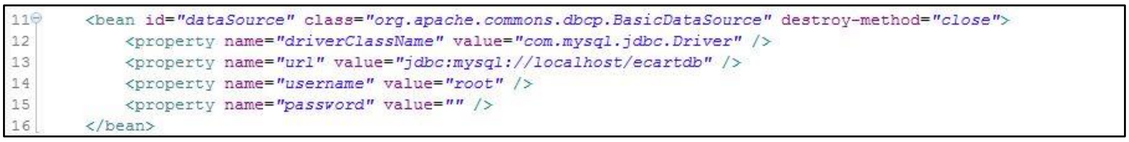
Hibernate ORM enables developers to more easily write applications whose data outlives the application process. As an Object/Relational Mapping (ORM) framework, Hibernate is concerned with data persistence as it applies to relational databases (via JDBC). [6]

Examples for domain class or POJO class, hibernate integration and hibernate database configuration can be shown in the following figures Figure 4.6, Figure 4.7 and Figure 4.8.



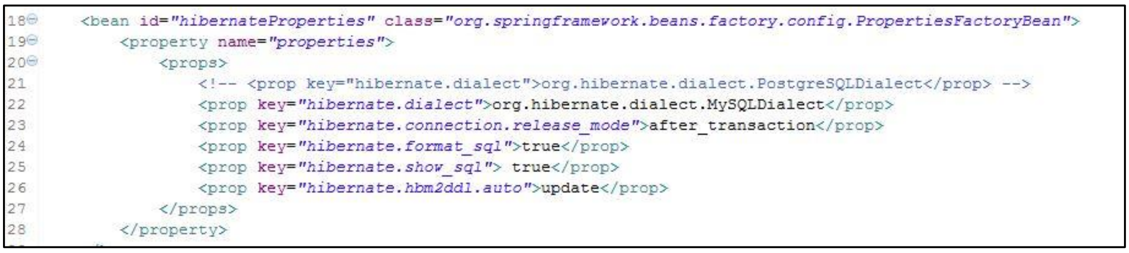
```java
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.TableGenerator;

import com.ecartlk.util.Trace;

@Entity
@Table(name = "PRODUCT")
public class Product extends Trace {
    private int productId;
    private int productCategoryId;
    private String name;
    private String code;

    @Id
    @GeneratedValue(strategy = GenerationType.TABLE, generator = "InvTab")
    @TableGenerator(name = "InvTab", table = "id_gen",
    pkColumnName = "id_name", valueColumnName = "id_val",
    allocationSize = 1, pkColumnValue = "Product")
    @Column(name = "PRODUCTID")
    public int getProductId() {
        return productId;
    }
    public void setProductId(int productId) {
        this.productId = productId;
    }

    @Column(name = "PROCATID")
    public int getProductCategoryId() {
```

Figure 4.6: Hibernate domain class



```xml
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
    <property name="driverClassName" value="com.mysql.jdbc.Driver" />
    <property name="url" value="jdbc:mysql://localhost/ecartdb" />
    <property name="username" value="root" />
    <property name="password" value="" />
</bean>
```

Figure 4.7: Hibernate database configuration



```
    <bean id="hibernateProperties" class="org.springframework.beans.factory.config.PropertiesFactoryBean">
        <property name="properties">
            <props>
                <!-- <prop key="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</prop> -->
                <prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
                <prop key="hibernate.connection.release_mode">after_transaction</prop>
                <prop key="hibernate.format_sql">true</prop>
                <prop key="hibernate.show_sql"> true</prop>
                <prop key="hibernate.hbm2ddl.auto">update</prop>
            </props>
        </property>
```

Figure 4.8: Hibernate configuration

4.7 Third party codes used in the system

Main target of a developing a software solution is to maximize the user friendliness file achieving the functional requirements. To increase the performance and the user friendliness, several existing codes were reused. Those third party libraries may be JavaScript or jar files.

Bootstrap

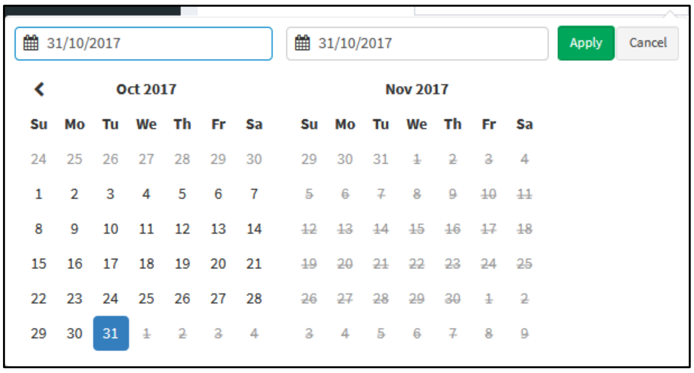
Whole website and the admin panel styles are developed on the bootstrap framework. Input fields, buttons, paginations, forms, icons, navigations, tabs and all other form elements are created based on design template in bootstrap.

Data Table

Datatable.js JavaScript plugin is used to generate grids in the system. This library gives many features such as search, paginations, sorts etc.

JQuery Date Picker

Date picker library is used to create attractive date pickers as well as date range pickers. Sample date range picker shown in the Figure 4.9



31/10/2017

31/10/2017

Oct 2017

Su	Mo	Tu	We	Th	Fr	Sa
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	~~1~~	~~2~~	~~3~~	~~4~~

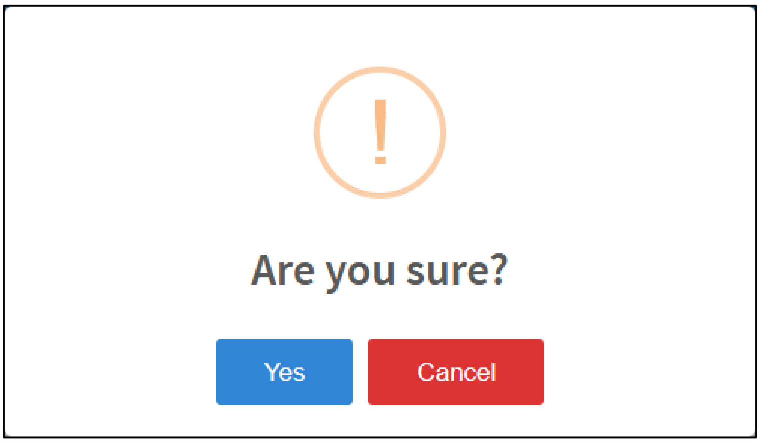
Nov 2017

Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	~~1~~	~~2~~	~~3~~	~~4~~
~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~
~~12~~	~~13~~	~~14~~	~~15~~	~~16~~	~~17~~	~~18~~
~~19~~	~~20~~	~~21~~	~~22~~	~~23~~	~~24~~	~~25~~
~~26~~	~~27~~	~~28~~	~~29~~	~~30~~	~~1~~	~~2~~
~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~

Figure 4.9: Date range picker

Popup messages and alerts

Sweetalert2.js library is used to create attractive and customized popups and alert messages. Figure 4.10 shown below in an alert message / confirmation message.



Are you sure?

Figure 4.10: Confirmation message

Chapter 05 – Evaluation

The main objective of this project is providing a useful software systems which helps client to do their work easily and with much efficient.

Evaluation phase is the process in software development cycle as it ensures that the product which is actually implemented matches with the requirements. Verification and validation is done during this phase to assure that the system meets its initial objectives, and testing should be planned at the very beginning of the project and carry out parallel with the development process.

5.1 Techniques of testing

A proper system testing should be carried out for evaluating the system to see its compatibility as a solution for the problem domain. Testing strategies may vary from product to another as software can fail in different ways unlike a physical product. It will lead the software to a poor quality state unless the testing is not done properly. Following Figure 5.1 shows the overview abstract idea of the testing.

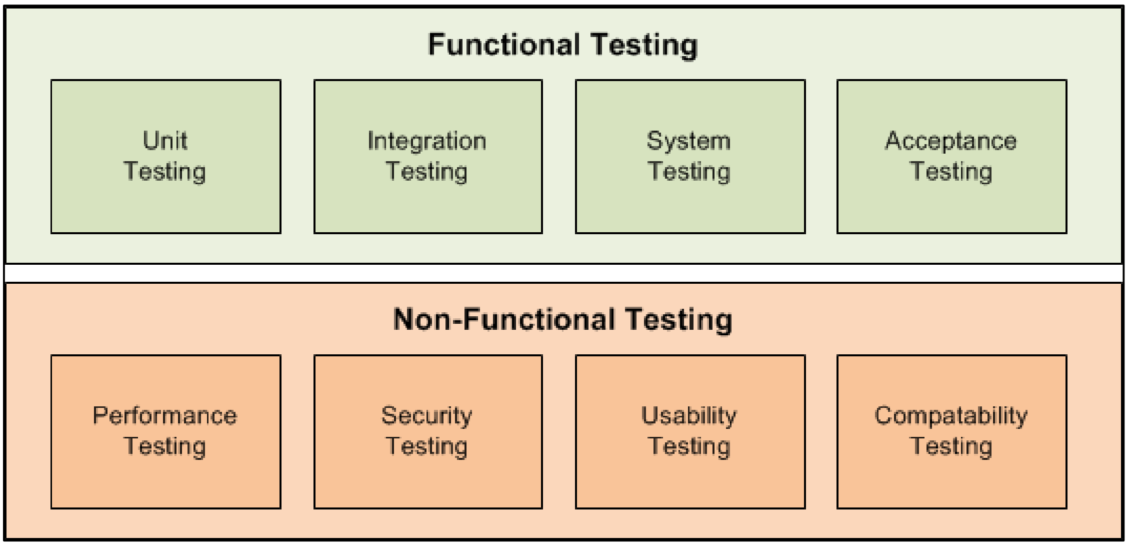


Figure 5.1: Functional and non-functional testing overview diagram. [7]

5.2 Test Plan for Proposed System

Testing plan of the system according to functional and non- functional testing are listed in the following sections.

5.2.1 Functional Testing

Functional testing of a developed system can be categorized into 4 major types as shown in the above Figure 5.1.

Unit Testing

Unit testing is very important as it can be used to find out errors within a module of a software. Each individual unit of user interfaces, classes should be tested for assuring their proper individual functionality.

Integration Testing

This testing is usually done with a combination of automated functional tests and manual testing. During integration testing it will check that the units tested in isolation works properly when they are put together.

System Testing

System as a whole should perform well concerning with all interdependencies. During System Testing it checks whether system works correctly when real clients typically use it. It will also checks how the system will perform if the system undergoes any extreme condition.

Acceptance Testing

System is tested at the client site using real data and check whether it works properly as expected.

5.2.2 Non-Functional Testing

Non-functional testing also very important same as functional testing. Non-functional testing is done against the non-functional requirement of the system.

Performance Testing

Performance testing or it can be named as load testing also. Performance testing ensure that the developed system is cable of handling load which are number of users and amount of data. Meanwhile stress testing can be done to find the failure points in the system when load testing is happening.

Security Testing

Security testing is a process intended to reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended. Due to the logical limitations of security testing, passing security testing is not an indication that no flaws exist or that the system adequately satisfies the security requirements. [8]

Security testing is consist of specific elements such as confidentiality, integrity, authentication, authorization, availability and non-repudiation.

Usability Testing

Easy to use the system and easy to navigate through the system.

Compatibility Testing

System should be work without any problem in recommended environments.

5.3 Test Cases

The test Plan of the system describes the testing strategies and the approaches to testing. By studying test cases of the system, it was discovered that the final system was satisfied user requirements and compatible with design specifications. Table 5.1 shows the test plan.

Module Name	**Function Name**	**Test Priority**
Login and User management	Login to the system (officer)	High
	Login to the website (Customer)	High
	Customer registration	High
	Officer registration	High
	Update profile	High
	Deactivate officer	High
	Log out from the system	High
	Log out from the website user account	High
Product management	Define product category	High

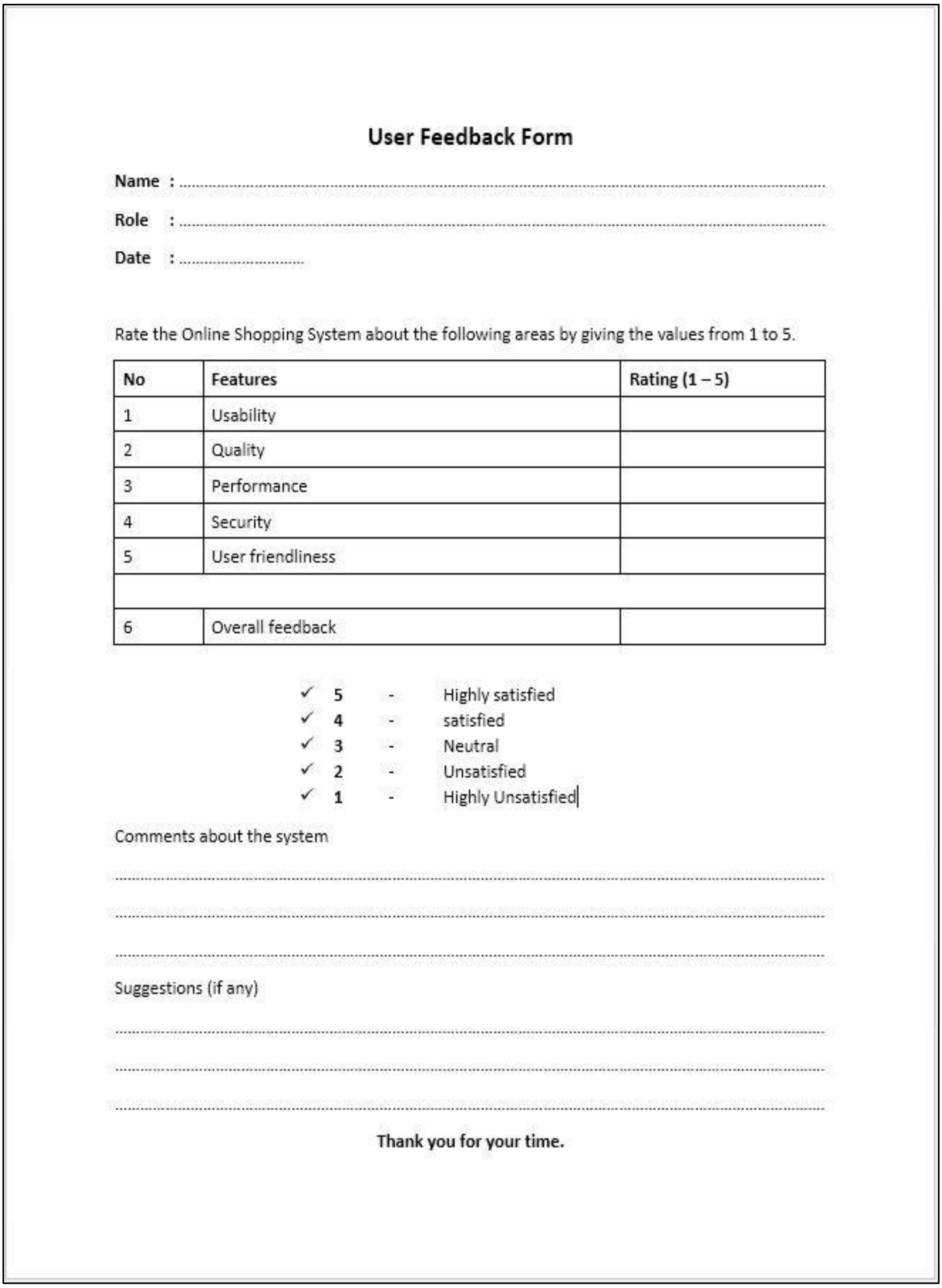
	Update / delete product category	High
	Define product	High
	Update / delete product	High
	Add items	High
	Update items	High
Inventory management	Inventory receive	High
	Assign inventory lot	High
	Generate report – inventory	High
Security management	Define access group	High
	Update / delete access group	High
	Define access control	High
	Update access control	High
	Access group mapping	High
Order management	View orders	Medium
	View order details (items)	Medium
	Update order status	High
	Assign couriers for orders	High
	Generate report regarding orders	High
Customer account handling	Add / update profile image	Low
	View order history	Medium
	Update order delivery	Medium
	Give feedback and rating	Medium
Purchase order	Search items	High
	View item	High
	Add item to cart	High
	View and update cart	High
	Checkout the cart	Very high

Report center	Order details	Medium
	Monthly transaction report	Very high
	Income and expenditure report	High
	Profit and loss report	High
	Most selling items report	High
	Stock reports	High

Table 5.1: High level test plan

5.4 User Evaluation

Getting user feedback is one of most important thing that should be done in the user acceptance test phase. In this process, user feedback forms are given to the system users and receive their feedback through the forms so that we can get a clear idea about the developed system from the user aspect. Following user feedback form in the Figure 5.2 shows the different aspect of the system that user must consider about and gives the rating from 1 to 5.



User Feedback Form

Name : ....

Role : ....

Date : ....

Rate the Online Shopping System about the following areas by giving the values from 1 to 5.

No	**Features**	**Rating (1 – 5)**
1	Usability	
2	Quality	
3	Performance	
4	Security	
5	User friendliness	
6	Overall feedback	

✓ **5** - Highly satisfied
✓ **4** - satisfied
✓ **3** - Neutral
✓ **2** - Unsatisfied
✓ **1** - Highly Unsatisfied

Comments about the system

....

....

....

Suggestions (if any)

....

....

....

Thank you for your time.

Figure 5.2: User feedback form

Chapter 06 – Conclusion

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur’s but also from the customer’s point of view. For the entrepreneurs, electronic shopping generates new business opportunities and for their customers, which makes comparative shopping possible. As per a survey, most customer of online stores are impulsive and usually make a decision to stay on a site within the first few seconds.

6.1 Encountered problems and lessons learnt

Facing problems in a software development is very common and rectify. Those problems measures the success of the project. Several problems were encountered during this project.

- Limited time period to do the development which was a main problem as there were vast research area to investigate.
- Increasing the performance of the system and the website.
- Enhancement in the client requirements were affected to the time frame of the project, which was again effected to the table structure of the system.

Main lesson learnt throughout this project is time management by delivering the product within a time frame. This project gave significant practical experience in software engineering disciplines. Most import part of software development life cycle is the analysis and design phase as I came to know by the practical experience. Learn about how to design a diagram using UML and develop the system using these diagrams. List of things learnt are as follows;

- Time management was an important fact to finish this project successfully.
- Experience of working with the client; how to manage them and how to respond to their requirements.
- Different business domain knowledge was gathered from this project in the requirement gathering phase.
- New technologies in programming languages and experience of working with new tools.

- Important of documentation and practical experience of making them was get to known.
- Opportunity to open up the mind to study more details of e-commerce systems and domain.
- Challenging and valuable project to enhance system development capability.
- A chance to strengthen the project management skills by providing deadlines to meet up and thereby heighten the time management skills. At the same time it enhanced the analyzing skills and given a unique opportunity to boost interpersonal skills by interact with the clients and to develop understanding on business philosophy.

6.2 Critical Assessment of the project

With compare to other similar systems, Ecart.lk is built with secure platform and easy to access. Ecart.lk provides items purchasing process with simple set of steps. It can be done with very few mouse clicks which is really easy to online customer. Unlike other similar systems, this provide the facility to carry out whole business functions other than online purchasing facility. Ecart.lk provides administrative portal for the internal users / officers to accelerate their business functions. Most of the similar projects unable to provide online admin portal as they have separate systems for those functions.

“Website design is like a shop interior. As an example, if the shop looks poor or similar to other shops, probably customer will pay attention to attractive shops. Hence it has designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible.

In this online shopping system project, the user is provided with an e-commerce web site which can be used to buy goods online. A good shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and ~~to~~ should be able to remove or add items to their cart.

The shopping cart function in this online shopping system, facilitates number of features that are designed to make the customer more comfortable. Finally the Ecart.lk Online Shopping System provides users a great opportunity to handle their work successfully and easily.

6.2.1 Comparison with existing similar projects

Following Table 6.1 shows the results gain at the comparison of the similar projects like the Ecart.lk.

	Ecart.lk	**Ebay.com**	**Mydeal.lk**
Online chatting for inquiry	Yes	Yes	No
Selling facility for online users	No	Yes	No
Search and filter	Yes	Yes	Yes
QR scan	Yes	Yes	No

Table 6.1: Comparison of similar projects

6.2.2 User evaluation summarizing

After taking user feedback we can summarized them. Analyzed and evaluated result of the feedback forms can be showed in graphical format as shown in the below Figure 6.1.

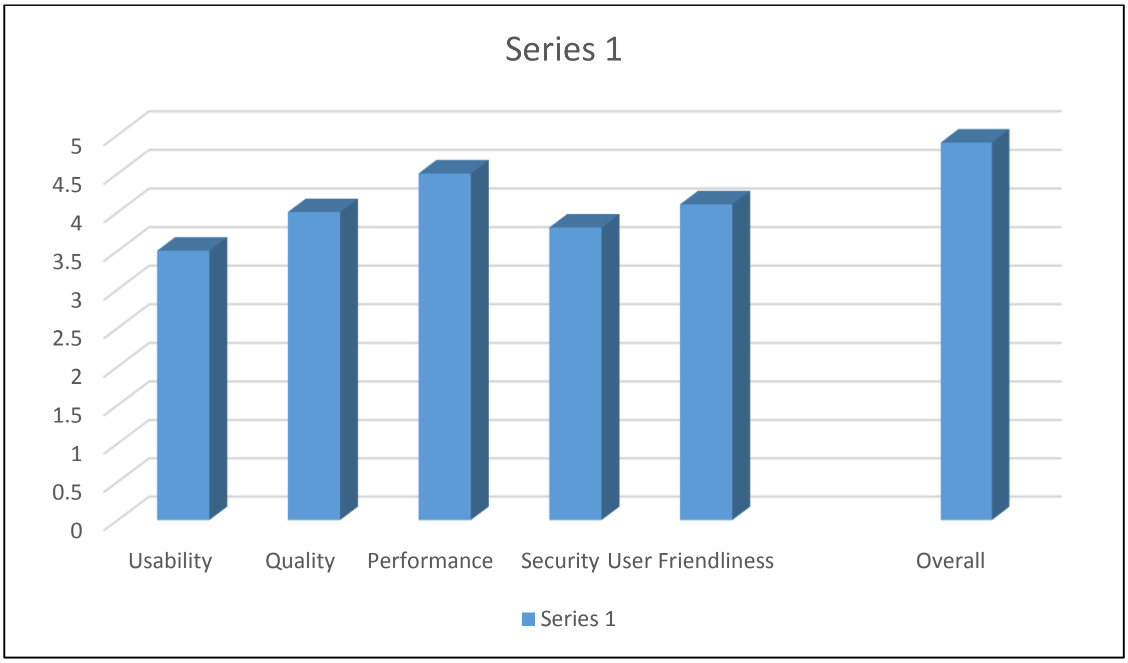


Figure 6.1: Summarized feedback value

6.3 Future developments

Some major client requests were arisen while the user acceptance testing. But those were not be able to complete with in this scope and the time frame. Hence they were agreed to continue few future developments as listed below.

- Online customer care service with a chatting module.
- Provide a total mobile application solution for the website of the system.
- Order tracking mechanism with a GPS facility which can be monitored the location and the courier of the order.
- Email and SMS notification facility for local customers about the special discounts and new arrival items.

References

[1] M. Rouse, "requirements analysis," [Online]. Available: http://searchsoftwarequality.techtarget.com/definition/requirements-analysis.

[2] K. Ewang, "Programming Methodologies," [Online]. Available: http://www.academia.edu/28464643/Programming_Methodologies.

[3] "Software User Interface Design," [Online]. Available: https://www.tutorialspoint.com/software_engineering/software_user_interface_design.htm.

[4] "Apache Struts," [Online]. Available: https://struts.apache.org.

[5] "Spring Tool Suite™," [Online]. Available: https://spring.io/tools/sts.

[6] "Hibernate ORM," [Online]. Available: http://hibernate.org/orm.

[7] "Testing Methodologies," [Online]. Available: https://www.inflectra.com/GraphicsViewer.aspx?url=~/Ideas/Topics/testing-methodologies.doc&name=wordml://03000001.png.

[8] "Security Testing," en.wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/Security_testing.

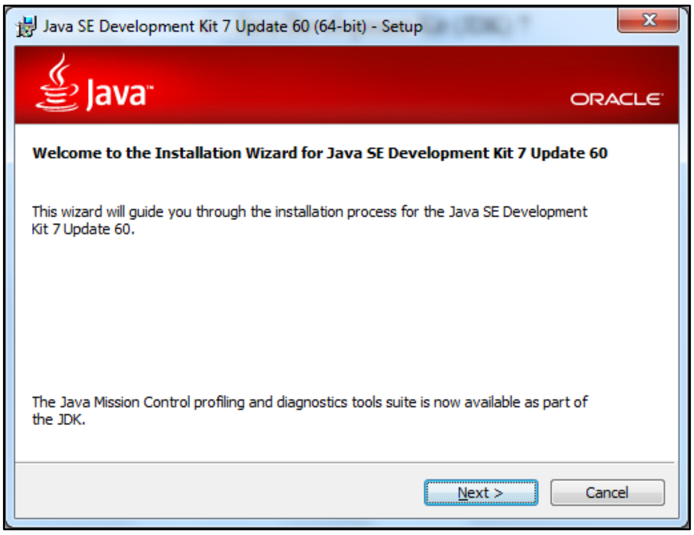
Appendix A - System Documentation

Establishing the system development environment must be done primarily in order to continue or do any improvements to the project. As mentioned in the section 4.1, list of software should be installed and configured properly.

- Java Development Kit (JDK) 7
- Apache tomcat 7.0.23
- MySQL server 5.6

Installing the JDK

- Java is open source programming language which can be downloaded freely from the website URL http://java.com.
- Download the relevant version to the computer.
- Run the downloaded file to open the java setup which is in the Figure A.1 and Figure A.2.

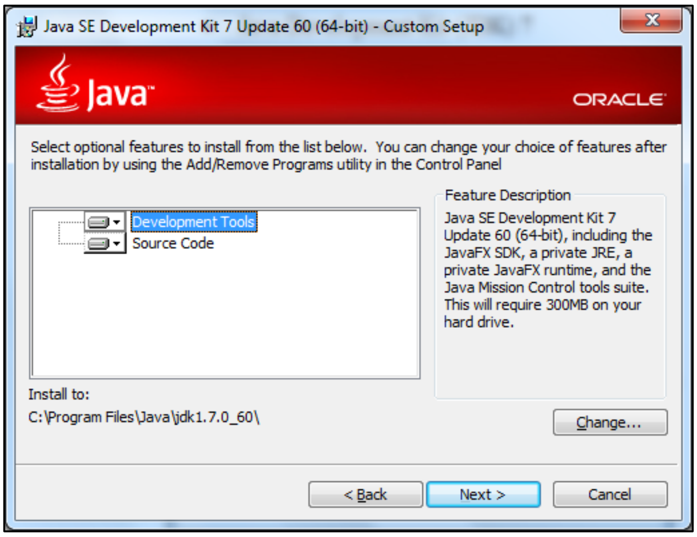


Welcome to the Installation Wizard for Java SE Development Kit 7 Update 60

This wizard will guide you through the installation process for the Java SE Development Kit 7 Update 60.

The Java Mission Control profiling and diagnostics tools suite is now available as part of the JDK.

Figure A.1: Java setup screen 1



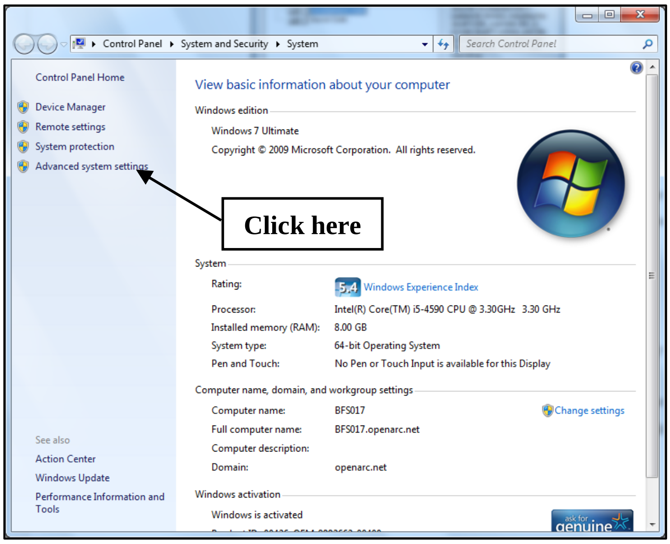
Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

Java SE Development Kit 7 Update 60 (64-bit), including the JavaFX SDK, a private JRE, a private JavaFX runtime, and the Java Mission Control tools suite. This will require 300MB on your hard drive.

Install to:
C:\Program Files\Java\jdk1.7.0_60\

Figure A.2: Java setup screen 2

- Click the next button in the above screens. JDK will install to the C drive of your computer by default. If you want to change the installation location you can change it in the screen which is Figure A.2.
- After that JRE installation screen will automatically appear.
- We can remain it to install into default location or change.
- After finish that Java installed to the computer successfully.
- Then JDK should be configured in the environment variable to run Java in any location of the computer.
- Right click on the MyComputer and go to properties or open the window in the location Control Panel → System and Security → System. Then following window will appear as in the Figure A.3.



View basic information about your computer

Windows edition

Windows 7 Ultimate

Copyright © 2009 Microsoft Corporation. All rights reserved.

System

Rating:	5.4 Windows Experience Index
Processor:	Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz 3.30 GHz
Installed memory (RAM):	8.00 GB
System type:	64-bit Operating System
Pen and Touch:	No Pen or Touch Input is available for this Display

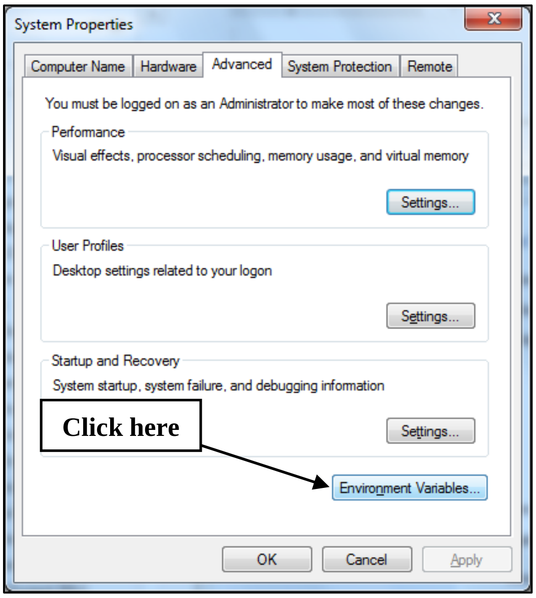
Computer name, domain, and workgroup settings

Computer name:	BFS017
Full computer name:	BFS017.openarc.net
Computer description:	
Domain:	openarc.net

Windows activation

Windows is activated

Figure A.3: System screen



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

Desktop settings related to your logon

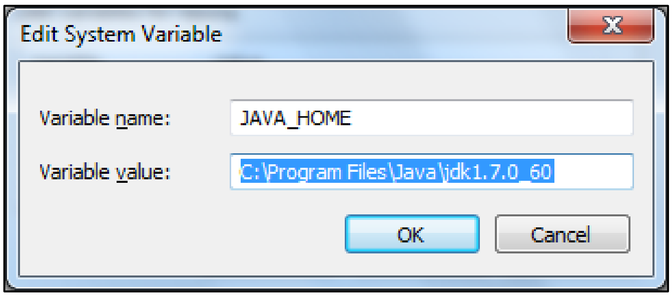
Startup and Recovery

System startup, system failure, and debugging information

Figure A.4: System properties screen

- Go to the environment variables screen as shown in the Figure A.4.

- Then create a new system variable named JAVA_HOME as in below Figure A.5.



Edit System Variable

Variable name: JAVA_HOME

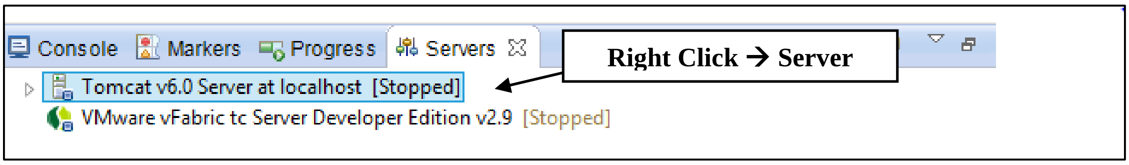
Variable value: C:\Program Files\Java\jdk1.7.0_60

Figure A.5: JAVA_HOME setup screen

- Then search for the Path variable and add the path of your JDK bin folder at the end of the variable value.

Installing the apache tomcat server

- Apache tomcat server which is also free can be downloaded from the URL http://tomcat.apache.org/download-70.cgi.
- Open the STS and add the tomcat server as displayed in the Figure A.6.



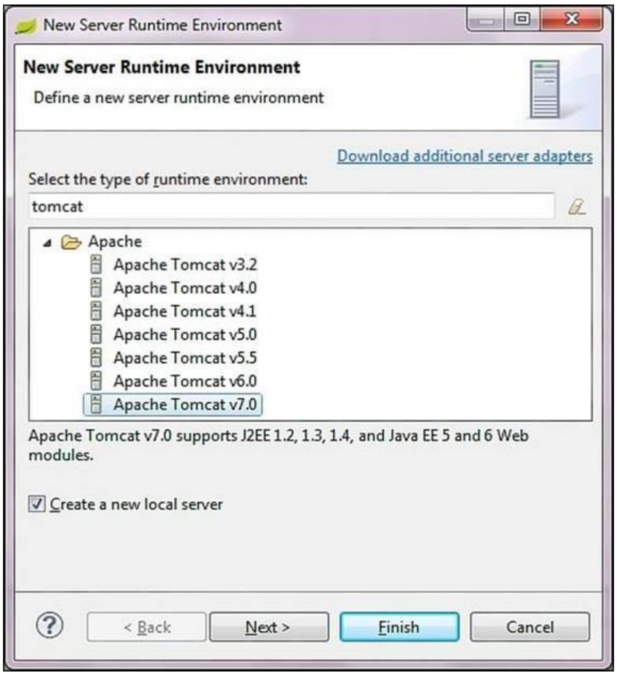
Right Click → Server

Tomcat v6.0 Server at localhost [Stopped]

VMware vFabric tc Server Developer Edition v2.9 [Stopped]

Figure A.6: Add new server

- Select the tomcat version as in the Figure A.7 and add the server path which is downloaded and the finish.



New Server Runtime Environment

New Server Runtime Environment

Define a new server runtime environment

Select the type of runtime environment: tomcat

- Apache
  - Apache Tomcat v3.2
  - Apache Tomcat v4.0
  - Apache Tomcat v4.1
  - Apache Tomcat v5.0
  - Apache Tomcat v5.5
  - Apache Tomcat v6.0
  - Apache Tomcat v7.0

Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.

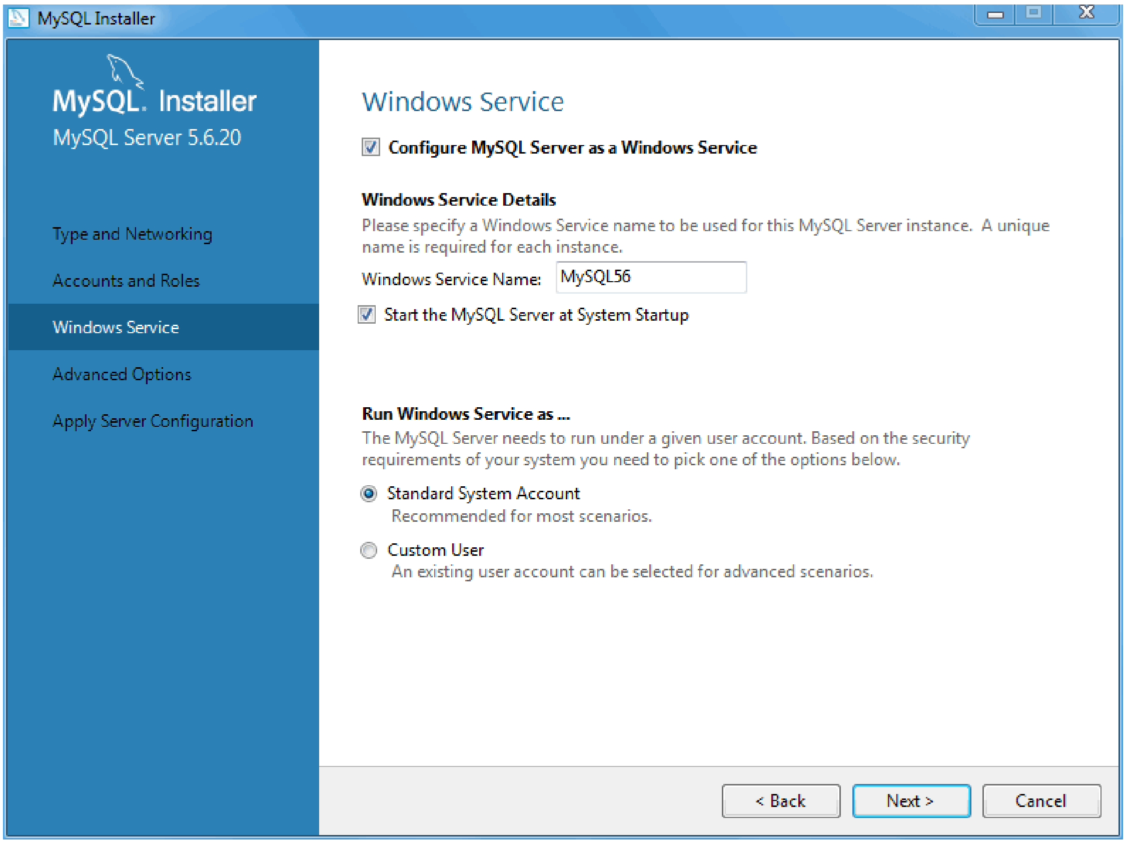
☑ Create a new local server

Figure A.7: Select server version

- Now double click the installed tomcat server and open the configuration.
- Select the server location.
- The open the launch configuration and add the following code segment to the VM argument to increase the memory.
  -Xmx1024m -XX:MaxPermSize=512m
- Apply the changes and click ok button.
- Tomcat HTTP port can be change if it is necessary.
- Save changes and select the server and start the server.

MySQL database installation

- MySQL Server 5.6 can be downloaded from using this URL https://dev.mysql.com. Select the correct version and download.
- Double click the setup file which is downloaded to installation and click next button on following screen at Figure A.8 to proceed.



Windows Service

☑ Configure MySQL Server as a Windows Service

Windows Service Details

Please specify a Windows Service name to be used for this MySQL Server instance. A unique name is required for each instance.

Windows Service Name: MySQL56

☑ Start the MySQL Server at System Startup

Run Windows Service as ...

The MySQL Server needs to run under a given user account. Based on the security requirements of your system you need to pick one of the options below.

● Standard System Account
Recommended for most scenarios.

○ Custom User
An existing user account can be selected for advanced scenarios.

Figure A.8: MySQL installer

- Select the program features you want to install from the setup screens.
- Select the installation location and click install button and then click finish.
- Start the MySQL server instance configuration wizard and click next.
- Select detailed configuration and click next.
- Finish the installation wizard and to view and restore the database backup file, Navicat Premium should be installation. Database can be restore using MySQl administrator too.

Navicat Premium Installation

- Download Navicat from the URL https://www.navicat.com/en/products.
- Go through the simple installation process and add connection after selecting the server type MySQL server.
- Then restore the SQL backup file.

Setup Ecart.lk Online Shopping System

- Open the STS and copy the EcartLK.rar file to the workplace.
- Unzip the source file.
- Import the project source code by file → import → browse file and select the project source that you have copied.
- Click next button and finish.
- Open the pom.xml file and set the deploy folder value to the webapps folder location of your installed tomcat server.
- Then open the spring-main.xml file and set the correct database name, database username and password according to the previous setups.

Run the Ecart.LK Online Shopping System

- Right click on the imported project folder in the package explorer and run as Maven Install.
- After project build is success then start the tomcat server.
- Finally developed system can be access through the URL http://localhost:8080/ecartlk. In the URL, localhost is the tomcat server location. If it in a separate server you can use its IP address. Then 8080 mean the port you have in the tomcat server installation.

Appendix B - Design Documentation

Diagrams and tables which are discussed earlier in the main chapters are discussed here in detail and high level diagrams can be found in this section.

Use case diagrams and descriptions

- Following Figure B.1 represent the use case diagram for the purchasing item from the online website.

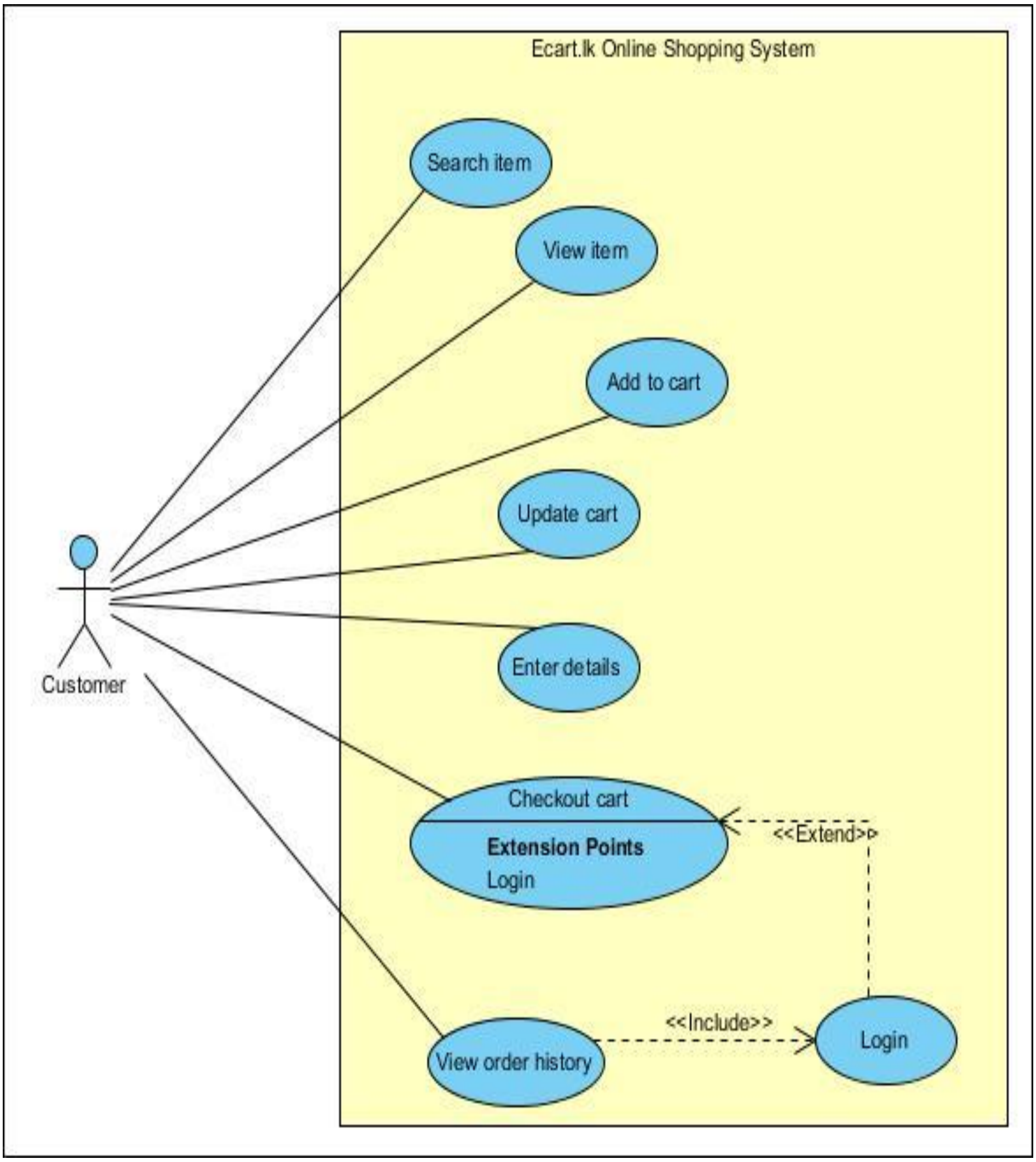


Figure B.1: Item purchase use case

According to above diagram, below Table B.1 describe the description of the purchasing item use case.

Use case	Purchasing items
Actors	Online customer
Overview	Order checkout by customer
Pre-conditions	1. Customer logged in to the system or 2. Can be checkout as guest user
Post-conditions	3. Customer can check out
Flow of events	
Actor action	**System response**
Update items in the cart	Change amount according to the quantity
Enter details and checkout	Confirmation message and reference number appear as successful message.

Table B.1: Item purchase use case description

- Use case diagram for adding items to the system can be find in the Figure B.2 shown below.

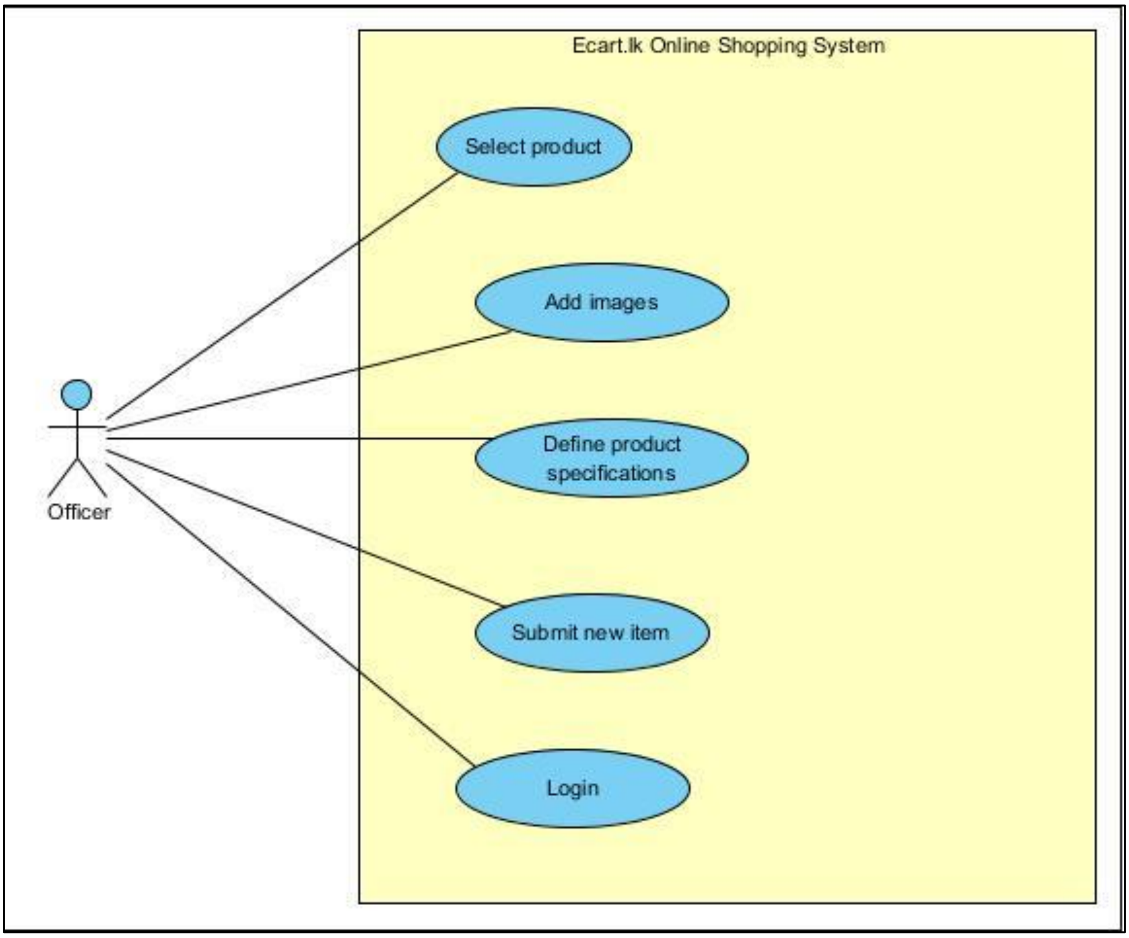

Ecart.lk Online Shopping System
Select product
Add images
Define product specifications
Submit new item
Login
Officer

Figure B.2: Use case for adding item

Table B.2 shows the description for the use case diagram which Figure B.2

Use case	Adding items
Actors	Officer
Overview	Adding items which can be done by officer.
Pre-conditions	1 Authorized officer should be logged in to system.
Post-conditions	2. Officer can fill item details and submit.
Flow of events	
Actor action	**System response**

Click “CREATE” button	Enable the form which use to enter details.
Enter all required fields	If all the required and correct format of values are entered, then it allow to submit
Submit new item	Response with successful message.
Alternative Flow of Event	
If the item is already defined, then error message will display on the screen.	

Table B.2: Adding item use case description

- Use case diagram for granting access rights for the system users shown in the below Figure B.3.

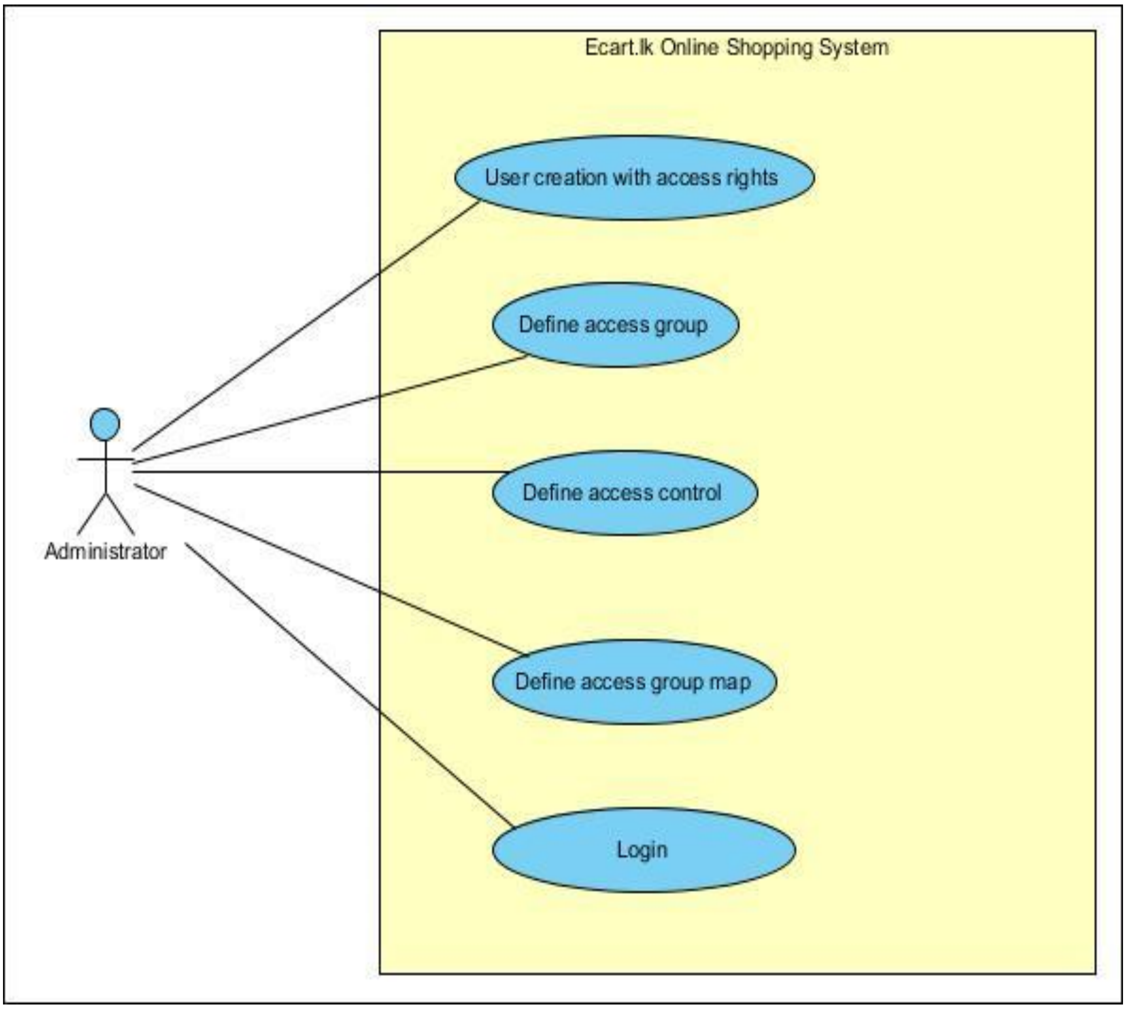


Figure B.3: Use case for granting access rights

Below Table B.3 represent the description of the use case of granting access rights to the system users.

Use case	Granting access rights to system users
Actors	Administrator
Overview	Access rights define by administrator
Pre-conditions	1. Must be logged in to the system
Post-conditions	2. User must be an administrator
Flow of events	
Actor action	**System response**
Define access group	Success message display on the screen if it is not available.
Define controls relevant to access groups	Display success message and access rights are granted to access group
Select access group in the user creation	Created user is mapped with the user group and display success message
Alternative Flow of Event	
Access group can be updated in the stakeholder profile update also.	

Table B.3: Access rights use case description

- Use case diagram for the assigning inventory in shown in the Figure B.4.

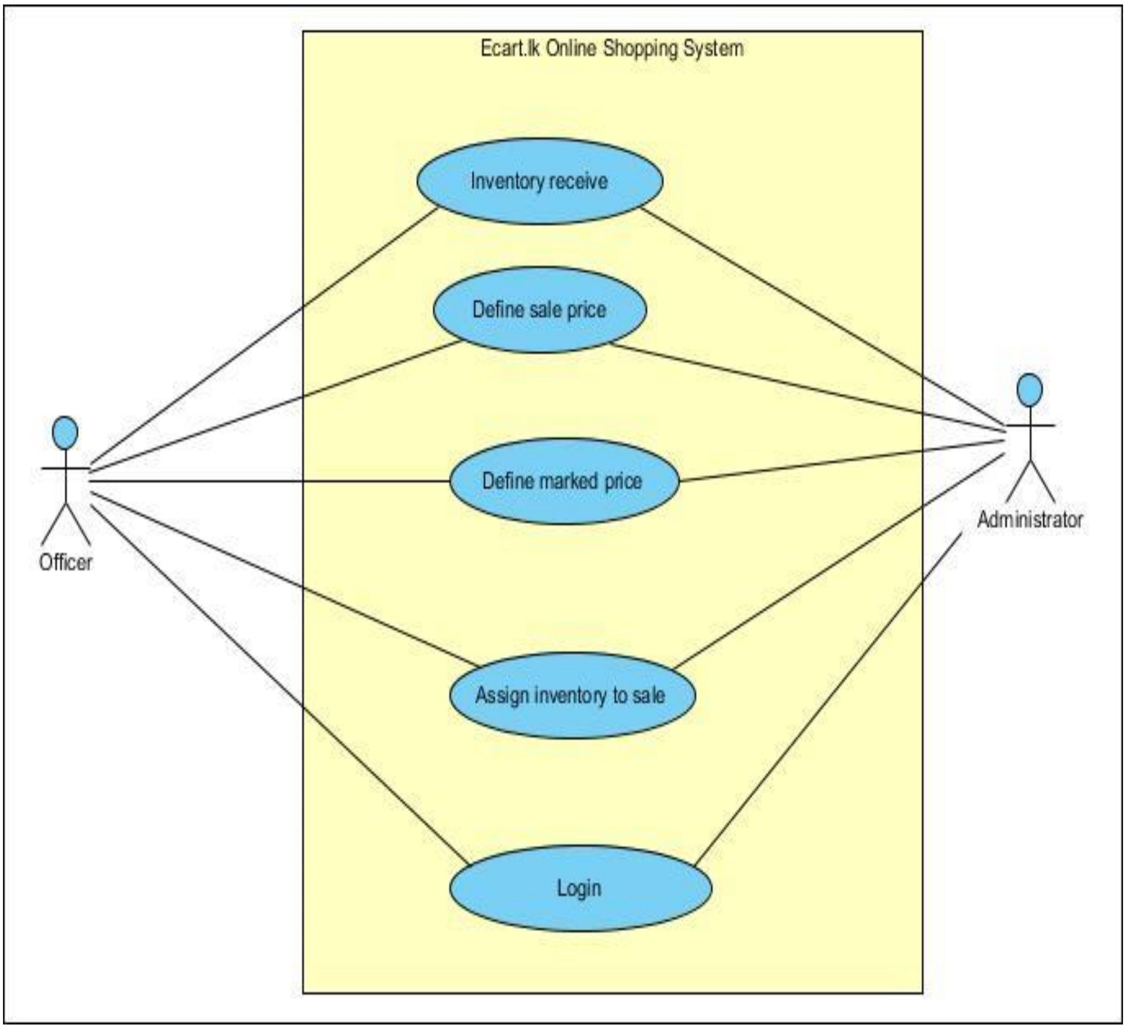


Figure B.4: Use case diagram for assigning inventory

Use case description for the above Figure B.4 is given in the below Table B.4.

Use case	Assign inventory to sale
Actors	Officer
Overview	Define sale price and assign inventory to be sale
Pre-conditions	1. Inventory must be receive
Post-conditions	2. Item will be ready to purchase in the website
Flow of events	

Actor action	System response
Select the received item from the drop down and click button search	Relevant records will loaded to the grid
Define sale price and marked price	Calculate profit from the sale price
Submit the record	Display the success message and items will display with the sale price in the website

Table B.4: Assign inventory use case description

Activity Diagram

Other than overall use case diagram which is shown in the design chapter, in this section detailed and high level use case diagrams are shown below.

- The Figure B.5 shows the activity diagram for the assign couriers.

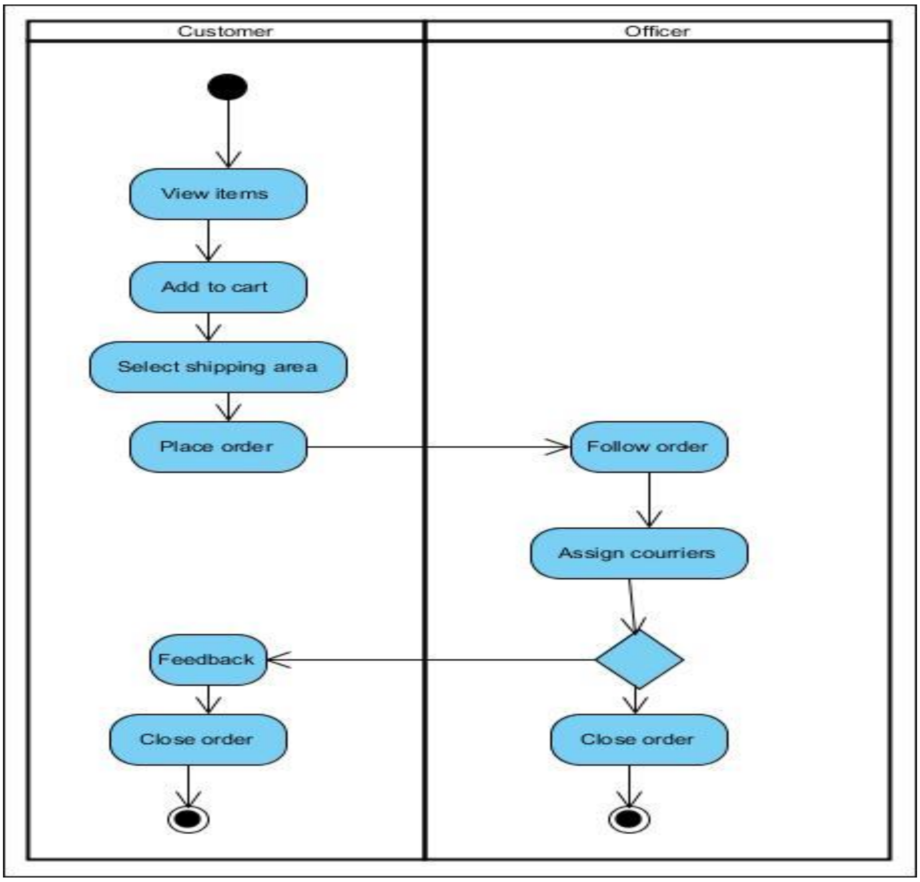


Figure B.5: Activity diagram for assign courier

- The Figure B.6 shows the activity diagram for the discount management.

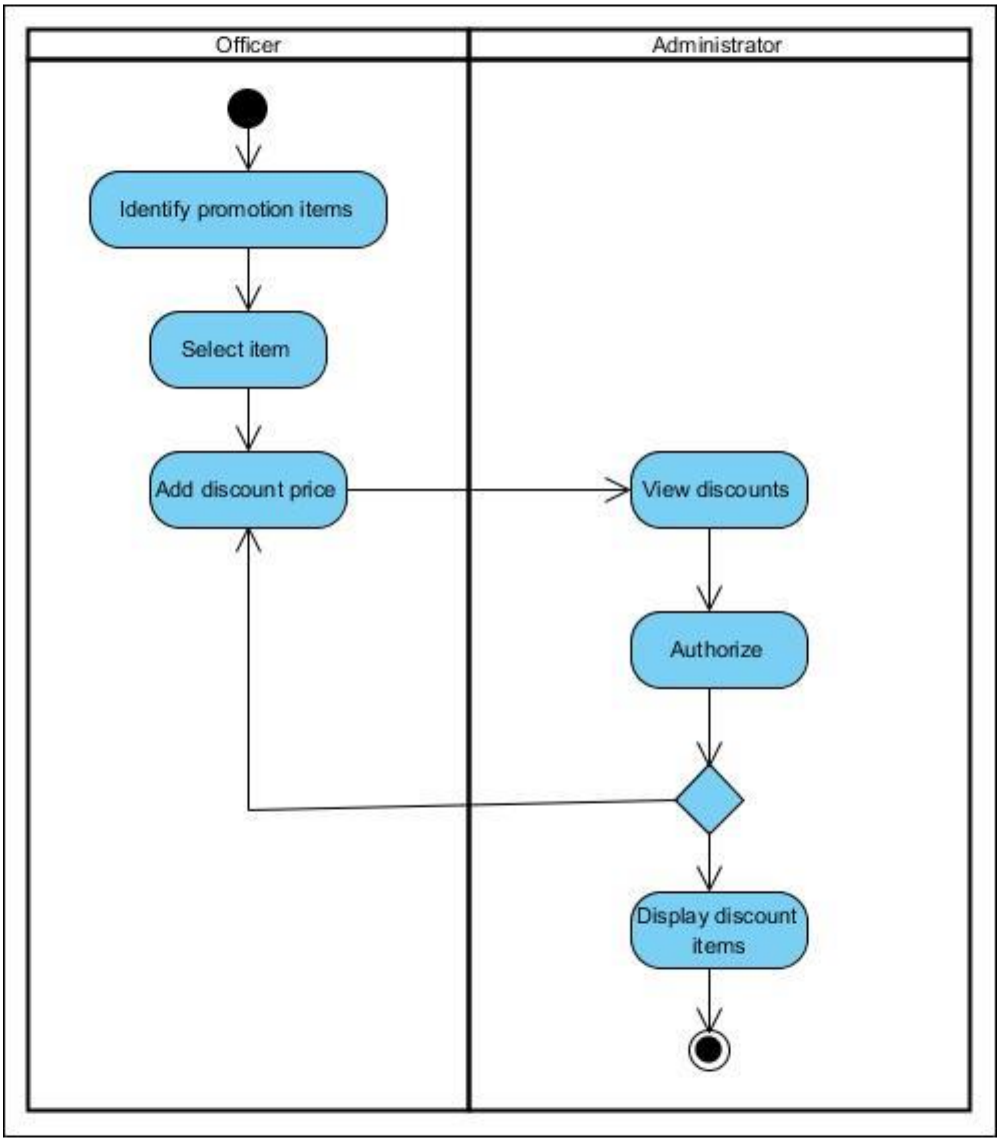
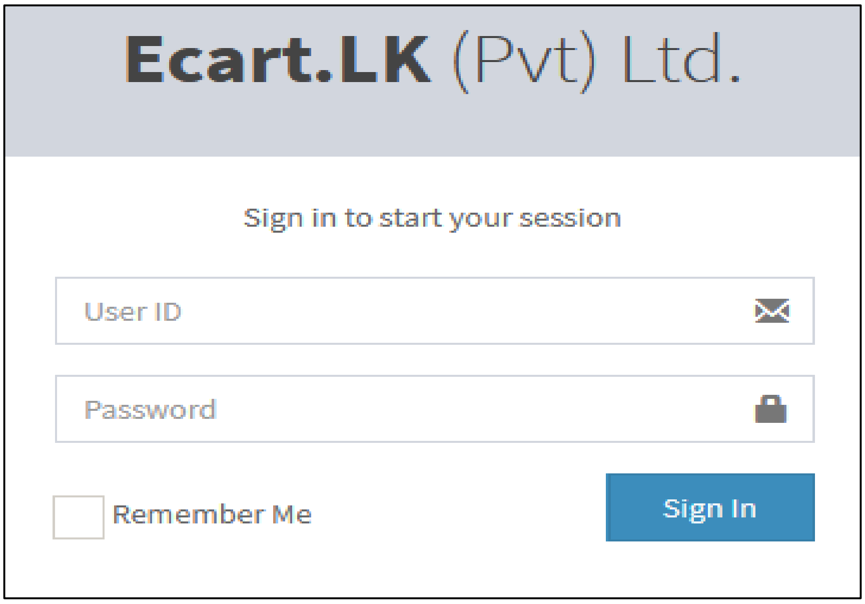


Figure B.6: Use case diagram for add discount

Appendix C - User Documentation

As mentioned in earlier chapters, this Ecart.lk Online Shipping System is consist of two major modules. That is the system and the website. System contain different and unique functionalities that can be access by different levels of privilege users. Privilege users can be managed in the system by the administrator.

First and foremost interface accessing every system users in login which is made log in to the system as well as user privileges. Login interface is shown in the following Figure C.1.



Ecart.LK (Pvt) Ltd.

Sign in to start your session

User ID

Password

□ Remember Me

Sign In

Figure C.1: Login interface

System users are shown the Figure 3.5 in design chapter as the home page which is consist of sales summaries, orders and some user accessible information. System users tree menu is consist of all the modules and screens. According to the access privileges, tree menu will show only the accessible user interfaces to the access groups. System tree menu with sub modules can be shown in the following Figure C.2

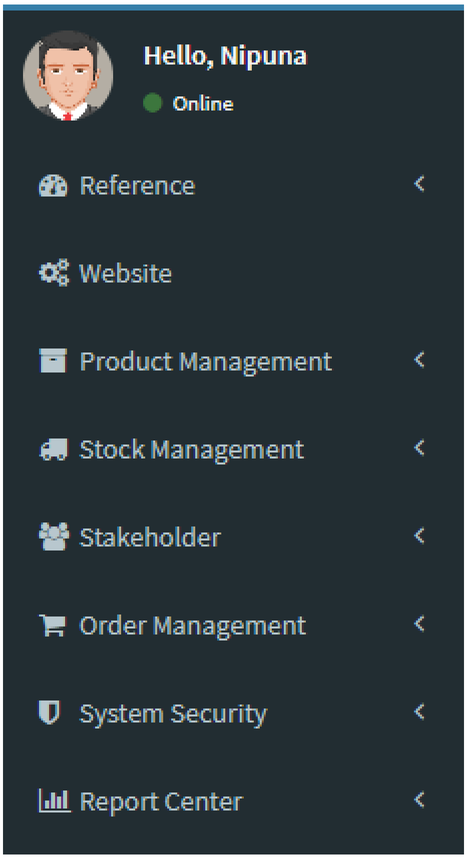


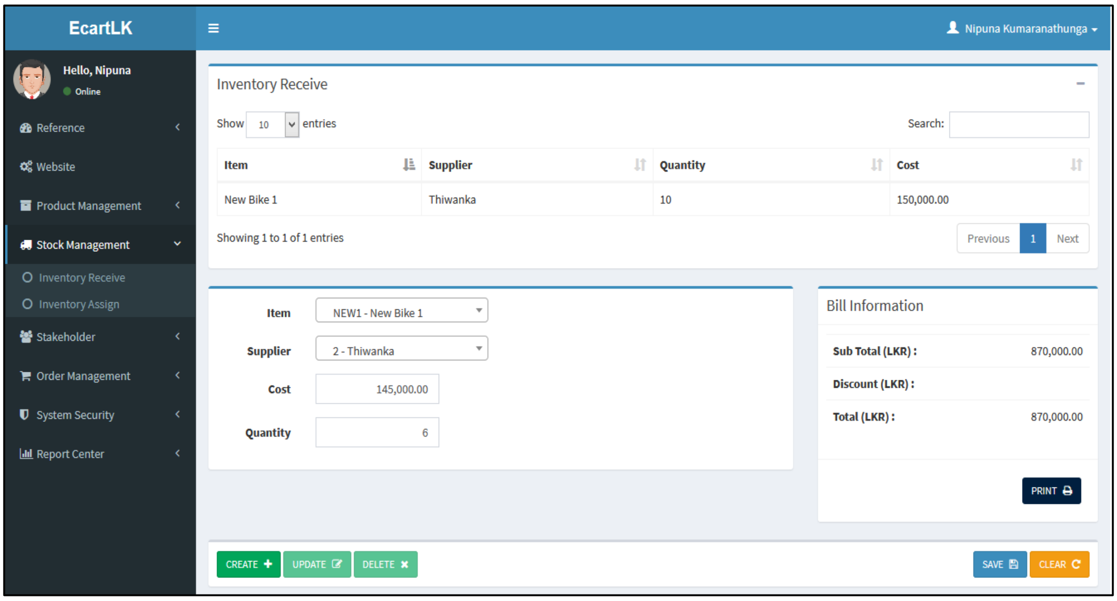
Figure C.2: System tree

Product Management

Product management section is consist with 4 sub menus such as product category, product, item and edit / view items. Product category and product sub menus are basically reference type screens that define categories and products. One of the most important sub menu in this product management menu is Item which is use to add items to the system and that can be shown in the Figure 3.7 in the chapter 3. Then the added items can be view and updated from the edit / view items section.

Inventory Management

Inventory management section has two parts inventory receive and inventory assign. Inventory receive UI records the inventories relevant to defined items. It keeps details of supplier details, cost and invoice can be generated to the supplier. Figure C.3 display inventory receive sub menu.



Inventory Receive

Show 10 entries Search:

Item	Supplier	Quantity	Cost
New Bike 1	Thiwanka	10	150,000.00

Showing 1 to 1 of 1 entries

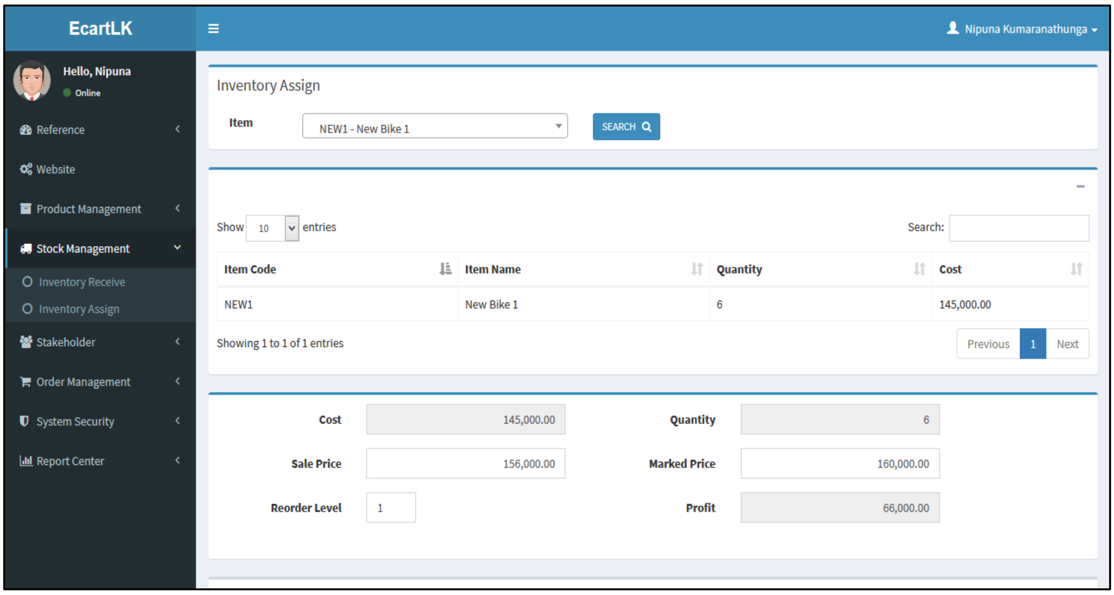
Item	NEW1 - New Bike 1
Supplier	2 - Thiwanka
Cost	145,000.00
Quantity	6

Bill Information

Sub Total (LKR) :	870,000.00
Discount (LKR) :	
Total (LKR) :	870,000.00

Figure C.3: Inventory receive screen

Inventory assign sub menu define the sale price and marked price for the items that are received from screen in Figure C.3. Figure C.4 shows the inventory assign UI.



Inventory Assign

Item NEW1 - New Bike 1

Show 10 entries Search:

Item Code	Item Name	Quantity	Cost
NEW1	New Bike 1	6	145,000.00

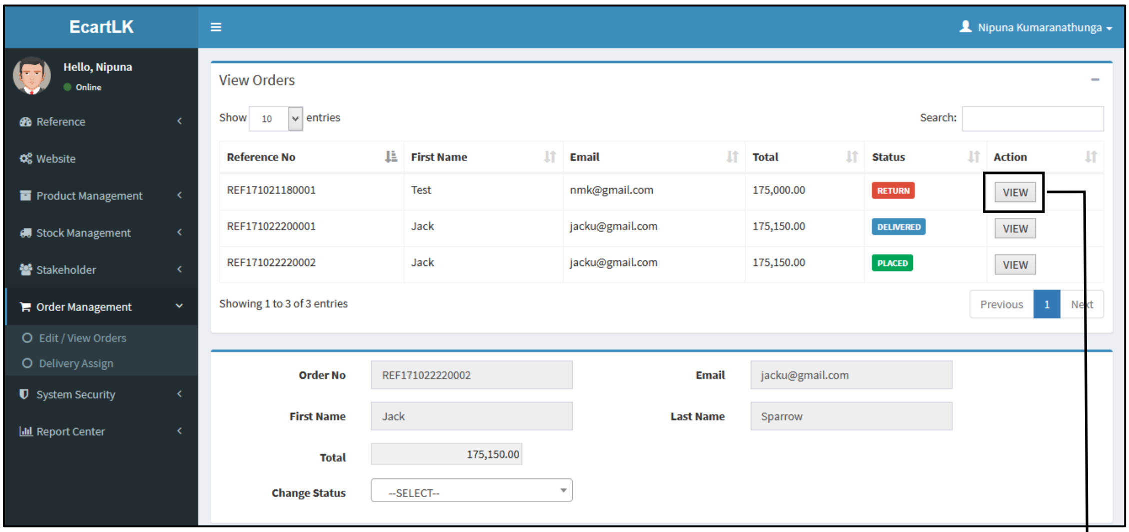
Showing 1 to 1 of 1 entries

Cost	145,000.00	Quantity	6
Sale Price	156,000.00	Marked Price	160,000.00
Reorder Level	1	Profit	66,000.00

Figure C.4: Inventory assign screen

Order Management

Order management section can be divided in to two parts Courier assign and Orders. View orders display order list with the reference numbers, status and the order details can be view in this sub section. Purchased items, quantity, discounts etc., can be view related to orders. Status of the orders also can be change. That can be shown in the following Figure C.5 and Figure C.6.



View Orders

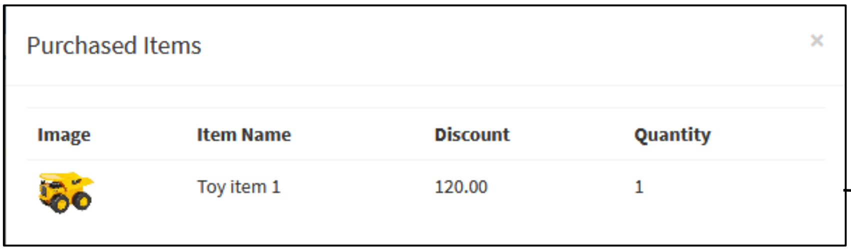
Show 10 entries　Search:

Reference No	First Name	Email	Total	Status	Action
REF171021180001	Test	nmk@gmail.com	175,000.00	RETURN	VIEW
REF171022200001	Jack	jacku@gmail.com	175,150.00	DELIVERED	VIEW
REF171022220002	Jack	jacku@gmail.com	175,150.00	PLACED	VIEW

Showing 1 to 3 of 3 entries　Previous 1 Next

Order No	REF171022220002	Email	jacku@gmail.com
First Name	Jack	Last Name	Sparrow
Total	175,150.00		
Change Status	--SELECT--		

Figure C.5: Orders screen

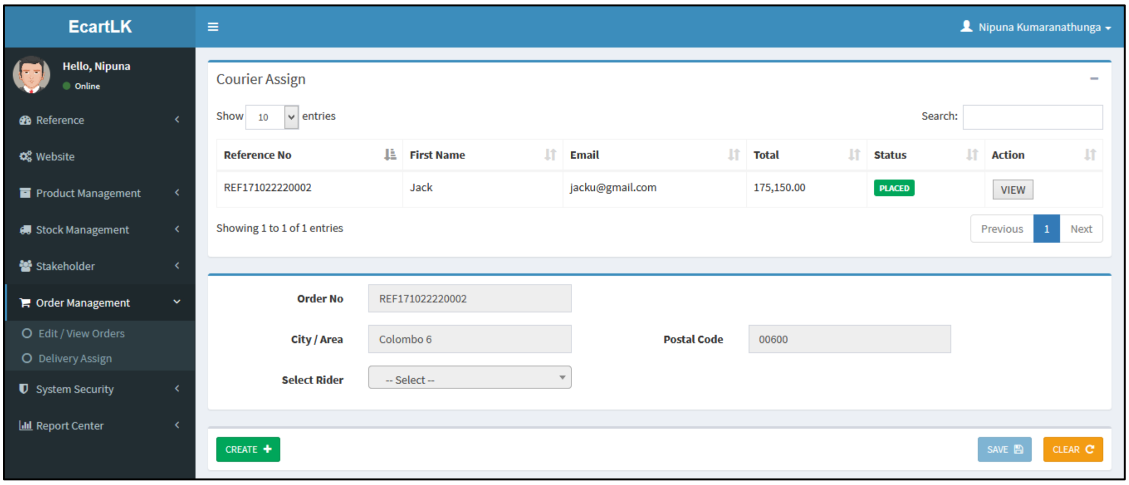


Purchased Items ×

Image	Item Name	Discount	Quantity
	Toy item 1	120.00	1

Figure C.6: Purchased items popup

Courier assign sub menu is used to assign couriers who registered in the system and after assignment they will receive an email or text message notifying with order details. Figure C.7 shows the screen.



Courier Assign

Show 10 entries　Search:

Reference No	First Name	Email	Total	Status	Action
REF171022220002	Jack	jacku@gmail.com	175,150.00	PLACED	VIEW

Showing 1 to 1 of 1 entries　Previous 1 Next

Order No	REF171022220002		
City / Area	Colombo 6	Postal Code	00600
Select Rider	-- Select --		

Figure C.7: Courier assign screen

Order Checkout

This is the most critical and most important function of an online shopping system. All the other system functionalities are focused on this process. Order checkout process is consist of few processes. First the customer can search any item from the website or he

/ she can browse through categories and sub categories. For marketing purposes, most viewed items, featured items, discount items, promotions can be seen in the home page of the website. Some of them can be shown in the Figure C.8.

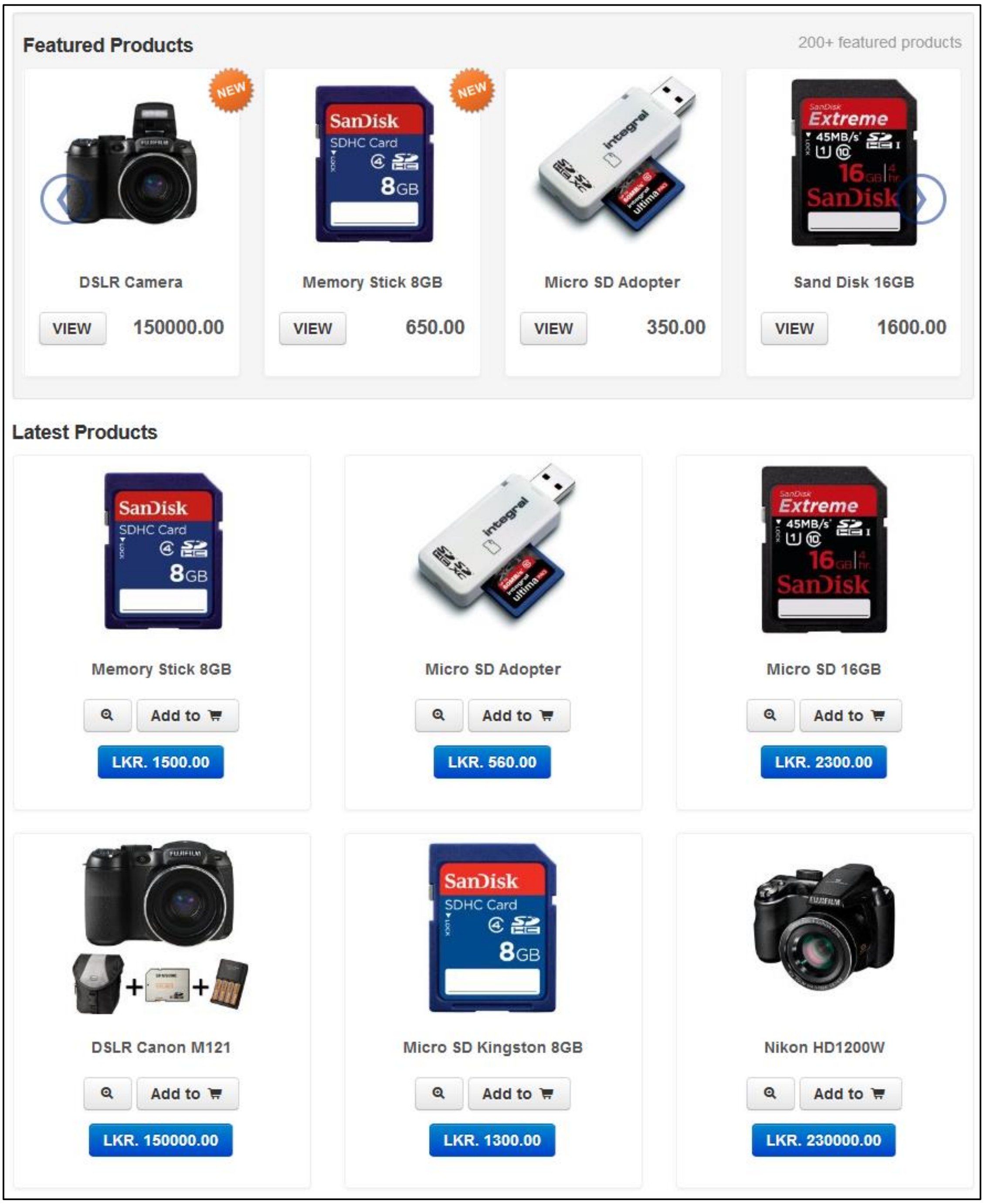
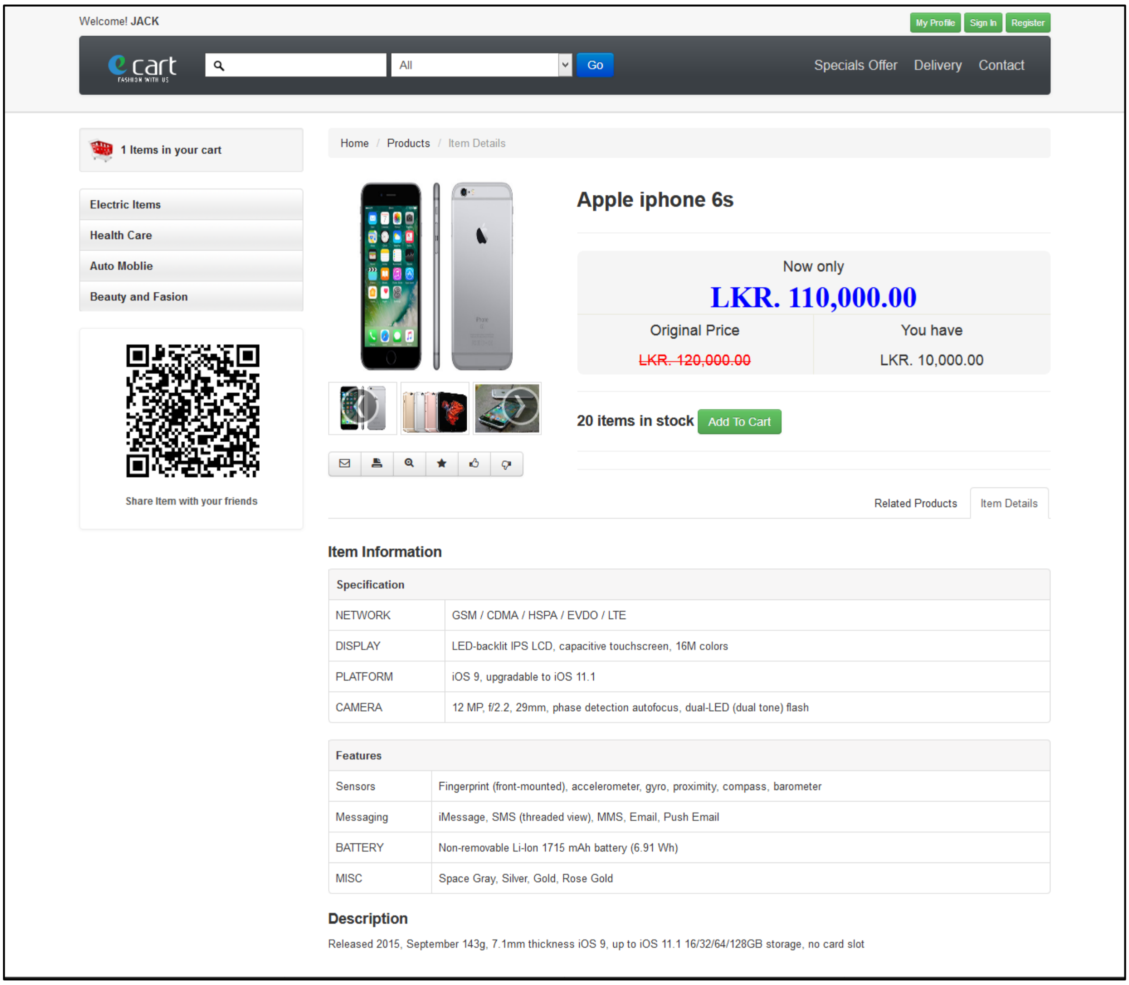


Figure C.8: Home page with items

Items can be view to see more details such as specification, features, images and descriptions. Item view screen is shown in the following Figure C.9.



Apple iphone 6s

Now only
LKR. 110,000.00

Original Price	You have
~~LKR. 120,000.00~~	LKR. 10,000.00

20 items in stock

Item Information

Specification	
NETWORK	GSM / CDMA / HSPA / EVDO / LTE
DISPLAY	LED-backlit IPS LCD, capacitive touchscreen, 16M colors
PLATFORM	iOS 9, upgradable to iOS 11.1
CAMERA	12 MP, f/2.2, 29mm, phase detection autofocus, dual-LED (dual tone) flash

Features	
Sensors	Fingerprint (front-mounted), accelerometer, gyro, proximity, compass, barometer
Messaging	iMessage, SMS (threaded view), MMS, Email, Push Email
BATTERY	Non-removable Li-Ion 1715 mAh battery (6.91 Wh)
MISC	Space Gray, Silver, Gold, Rose Gold

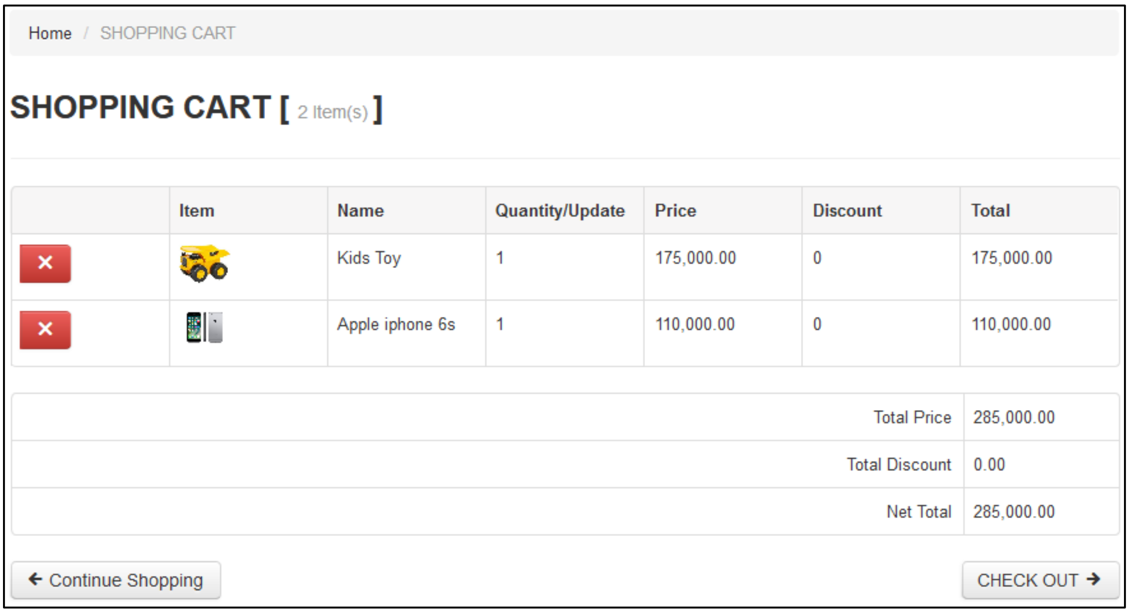
Description

Released 2015, September 143g, 7.1mm thickness iOS 9, up to iOS 11.1 16/32/64/128GB storage, no card slot

Figure C 9: Item view screen

Click on Add To Cart button to add item to the shopping cart. Customer can decide the quantity from cart or at the checkout screen. QR code that is in the left of the screen can be used to identify the item and customer can share the item with others or he / she can access the URL from the browser of a mobile device directly from scanning QR code.

After continue on the shopping all the added items can be viewed in the shopping cart. Customer can manage the quantity or remove items if he / she want from the shopping cart. Shopping cart UI can be shown in the Figure C.10.



Home / SHOPPING CART

SHOPPING CART [2 Item(s)]

	Item	Name	Quantity/Update	Price	Discount	Total
✕		Kids Toy	1	175,000.00	0	175,000.00
✕		Apple iphone 6s	1	110,000.00	0	110,000.00

Total Price	285,000.00
Total Discount	0.00
Net Total	285,000.00

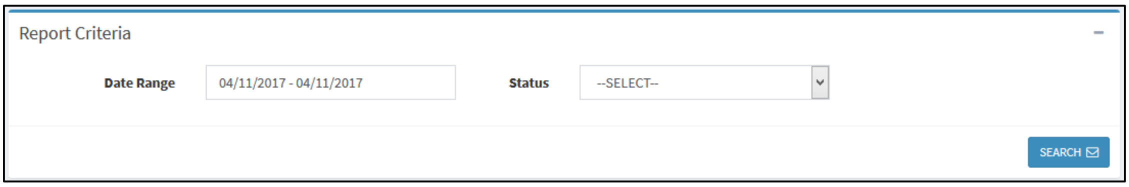
Figure C.10: Shopping cart screen

To finish the final stage of the order click on CHECK OUT ➔ button or if customer wish to continue on shopping click on ← Continue Shopping button.

Then the screen will appear as shown in the figure 3.6 in chapter 3 and customer can add relevant details and finish the checkout. Then the successful notification will appear on the screen and the order reference number will display in the screen.

Appendix D - Management Reports

Management reports generated from the system helps the company managers and all decision making parties to get company decisions. So that administrators and also officers should have ability generate details reports as well as summarized reports from the system. Reports can be generated from the Report Center of the system. Some of report criteria can be shown in the following Figure D.1



Report Criteria

Date Range: 04/11/2017 - 04/11/2017

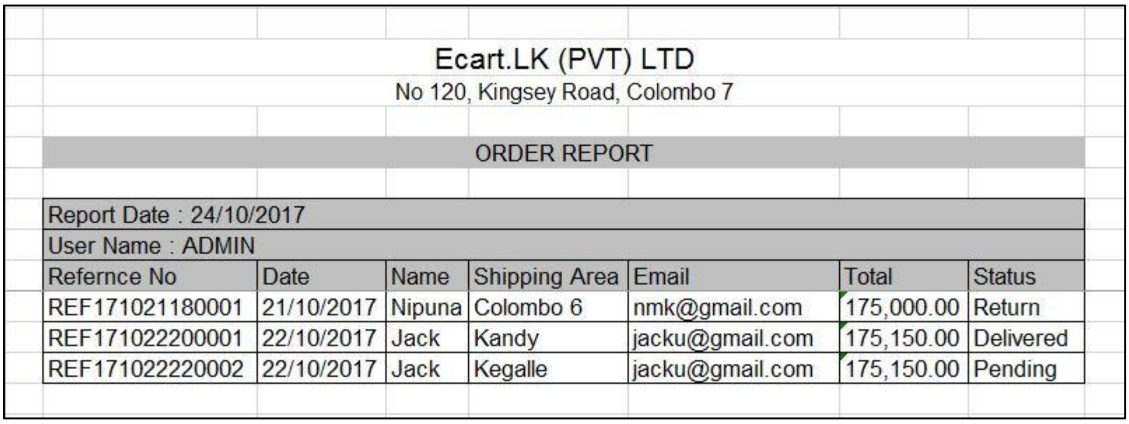
Status: --SELECT--

Figure D.1: Report criteria

Admin can select the date range or status (if it is available) to filter the reports that are generated. Reports can be generate in formats PDF or Excel.

Order Report

Order report is one of basic report that can be generated from the system. Report can be filtered from date range, order status, shipping area etc., Following Figure D.2 shows the order report of the system.



Ecart.LK (PVT) LTD

No 120, Kingsey Road, Colombo 7

ORDER REPORT

Report Date : 24/10/2017

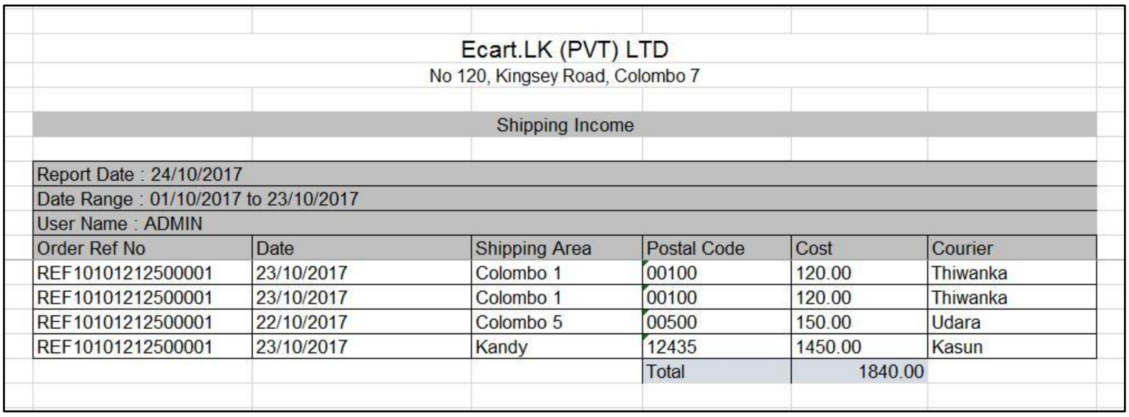
User Name : ADMIN

Refernce No	Date	Name	Shipping Area	Email	Total	Status
REF171021180001	21/10/2017	Nipuna	Colombo 6	nmk@gmail.com	175,000.00	Return
REF171022200001	22/10/2017	Jack	Kandy	jacku@gmail.com	175,150.00	Delivered
REF171022220002	22/10/2017	Jack	Kegalle	jacku@gmail.com	175,150.00	Pending

Figure D.2: Order report - Excel

Shipping Income Report

Most of the orders payment methods are cash on delivery. So that there is a shipping cost for most orders. In the Figure D.3 shows the shipping income report of the company. It also can be generated by filtering the result in a given date range or shipping area.



Ecart.LK (PVT) LTD
No 120, Kingsey Road, Colombo 7

Shipping Income

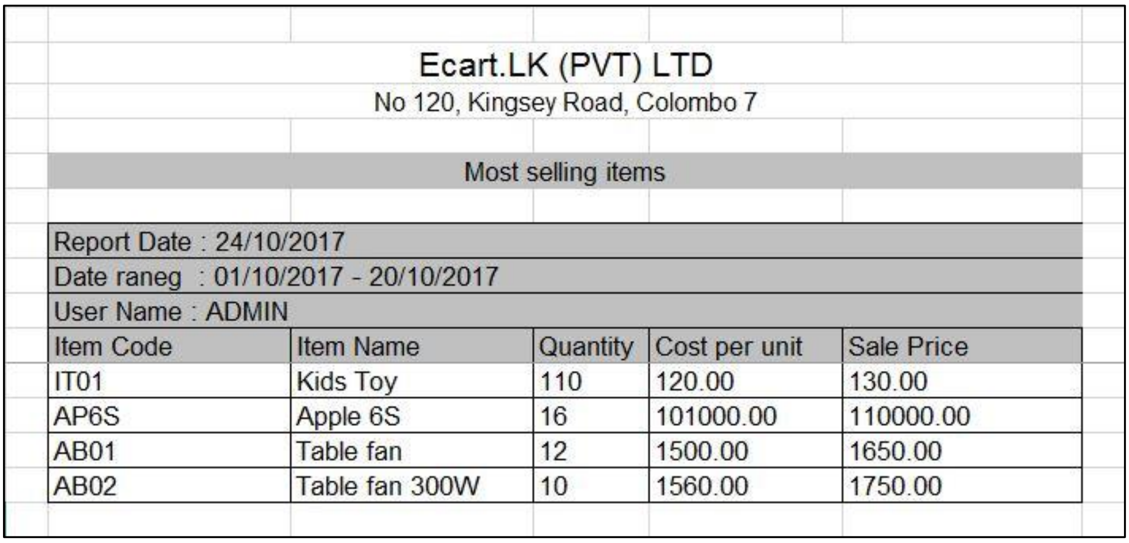
Report Date : 24/10/2017
Date Range : 01/10/2017 to 23/10/2017
User Name : ADMIN

Order Ref No	Date	Shipping Area	Postal Code	Cost	Courier
REF10101212500001	23/10/2017	Colombo 1	00100	120.00	Thiwanka
REF10101212500001	23/10/2017	Colombo 1	00100	120.00	Thiwanka
REF10101212500001	22/10/2017	Colombo 5	00500	150.00	Udara
REF10101212500001	23/10/2017	Kandy	12435	1450.00	Kasun
			Total	1840.00	

Figure D.3: Shipping income report - Excel

Most selling items

Most selling items in a given date range is also really important decision making report that can be generated. Because it will decide what items to purchase more in the future. This report contain the number of items sold according to different items with their sale prices. Figure D.4 shows the sample report.



Ecart.LK (PVT) LTD
No 120, Kingsey Road, Colombo 7

Most selling items

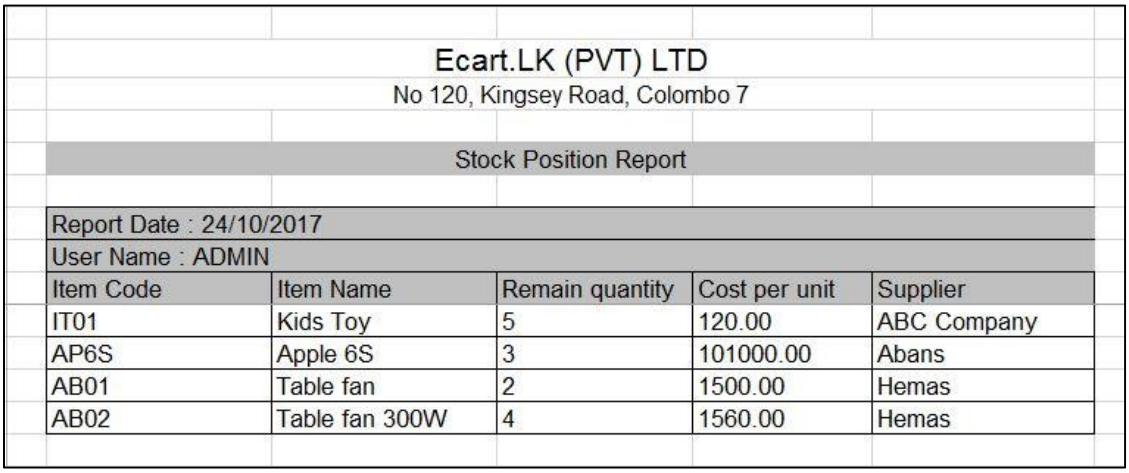
Report Date : 24/10/2017
Date raneg : 01/10/2017 - 20/10/2017
User Name : ADMIN

Item Code	Item Name	Quantity	Cost per unit	Sale Price
IT01	Kids Toy	110	120.00	130.00
AP6S	Apple 6S	16	101000.00	110000.00
AB01	Table fan	12	1500.00	1650.00
AB02	Table fan 300W	10	1560.00	1750.00

Figure D.4: Most selling items - Excel

Stock Position Report

This report contain the items that are need to be purchase as it those items were going on the low stock status. So that company can purchase more items to be sell. Officers or administrator can generate a filtered report by the supplier also. Figure D.5 represent the sample report of the stock position.



Ecart.LK (PVT) LTD
No 120, Kingsey Road, Colombo 7

Stock Position Report

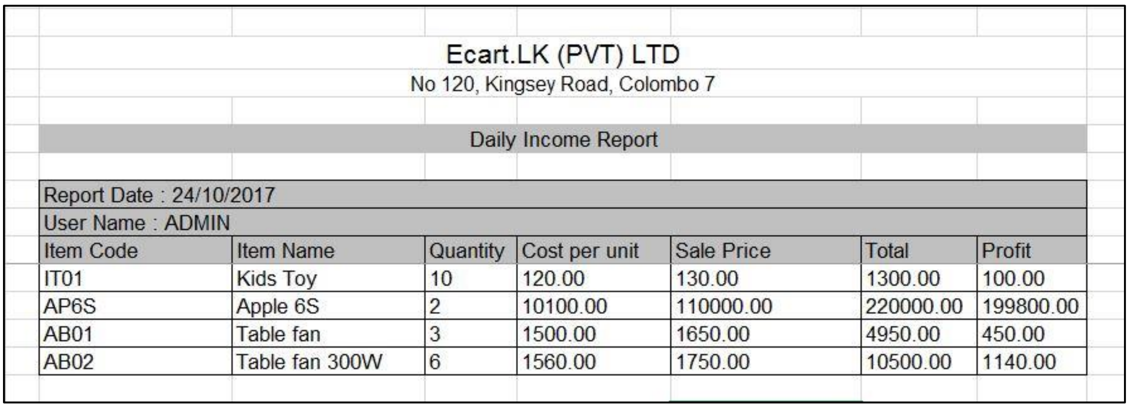
Report Date : 24/10/2017
User Name : ADMIN

Item Code	Item Name	Remain quantity	Cost per unit	Supplier
IT01	Kids Toy	5	120.00	ABC Company
AP6S	Apple 6S	3	101000.00	Abans
AB01	Table fan	2	1500.00	Hemas
AB02	Table fan 300W	4	1560.00	Hemas

Figure D.5: Stock position report - Excel

Income Report

Income report contain the details and information of transaction in given date range. It may contain number of items sold with the total cost and income. Below diagram D.6 shows the generated such report.



Ecart.LK (PVT) LTD
No 120, Kingsey Road, Colombo 7

Daily Income Report

Report Date : 24/10/2017
User Name : ADMIN

Item Code	Item Name	Quantity	Cost per unit	Sale Price	Total	Profit
IT01	Kids Toy	10	120.00	130.00	1300.00	100.00
AP6S	Apple 6S	2	10100.00	110000.00	220000.00	199800.00
AB01	Table fan	3	1500.00	1650.00	4950.00	450.00
AB02	Table fan 300W	6	1560.00	1750.00	10500.00	1140.00

Figure D.6: Income report - Excel

Appendix E - Test Results

According to the test plans mentioned in the Chapter 5, test results of the system are shown below in the tabulate form. Test results are generated by evaluating the scenarios with every combination of inputs.

P – Pass and F – Fail

Test case for the login is shown in the Table E.1.

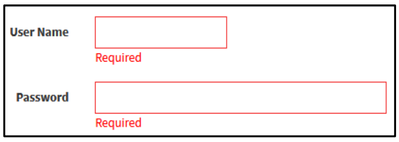
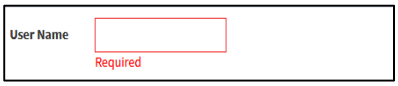
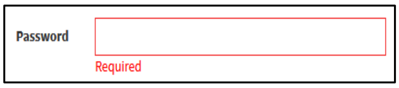
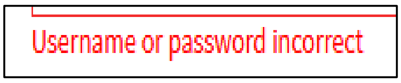
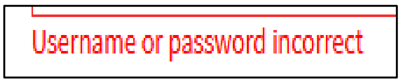
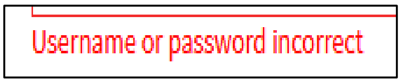
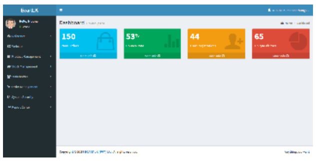
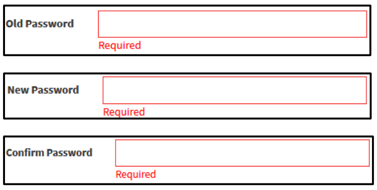
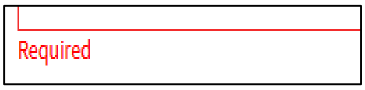
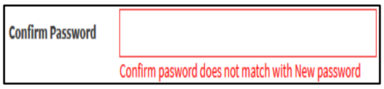
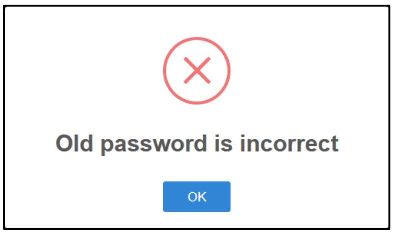
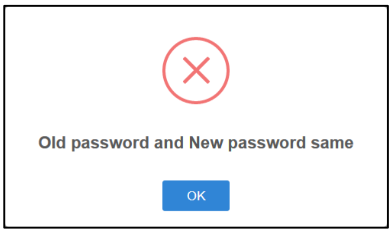
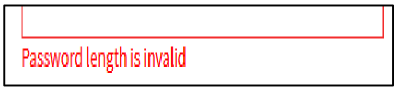
No	Test Case	Expected Output	Actual Output	P/F
1	Click login without entering username and password	Display error	 User Name Required Password Required	P
2	Click login without entering username	Display error	 User Name Required	P
3	Click login without entering password	Display error	 Password Required	P
4	Click login with correct username and wrong password	Display error	 Username or password incorrect	P
5	Click login with correct password and wrong username	Display error	 Username or password incorrect	P
6	Click login with both incorrect username and password	Display error	 Username or password incorrect	P
7	Click login with both correct username and password	Redirect to home page	 	P

Table E.1: Test result for Login

Test case for validate the change password in user profile is shown in the following Table E.2.

No	Test Case	Expected Output	Actual Output	P/F
1	Click Change password without entering old password, new password and confirm password	Display error	 Old Password Required New Password Required Confirm Password Required	P
2	Click Change password with any one field empty	Display error in all fields	 Required	P
3	Click Change password with all fields filled but new password and confirm password are different	Display error	 Confirm Password Confirm pasword does not match with New password	P
4	Click Change password with old password incorrect	Display error	 Old password is incorrect OK	P
5	Click Change password with old password and new password same	Display error	 Old password and New password same OK	P
6	Click Change password with new password less than 6 character in length	Display error	 Password length is invalid	P

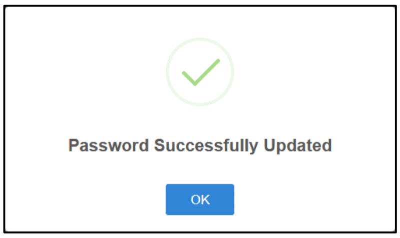
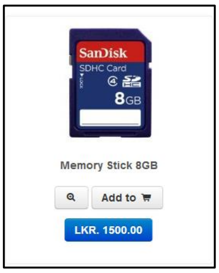
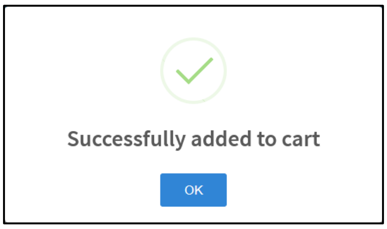
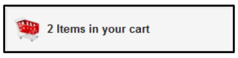
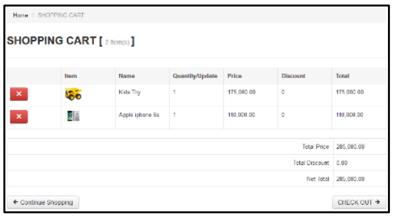
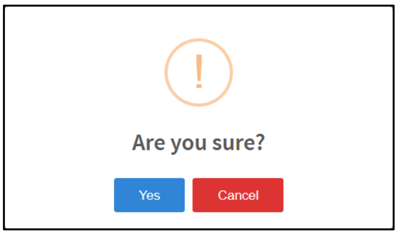
7	Click Change password with all correct values	Successful message display	 Password Successfully Updated OK	P

Table E.2: Test result for user profile change password

Test case for the process of checkout an order shown in the following Table E.3.

No	Test Case	Expected Output	Actual Output	P/F
1	Browse for the items in the website	Display the list of items in the website	 Memory Stick 8GB Add to LKR. 1500.00	P
2	Click on the add to cart button	Display success message	 Successfully added to cart OK	P
3	Click on the  2 Items in your cart	Display shopping cart	 Home / SHOPPING CART SHOPPING CART [2 Item(s)] Item | Name | Quantity/Update | Price | Discount | Total Kids Toy | 1 | 175,000.00 | 0 | 175,000.00 Apple iphone 6s | 1 | 110,000.00 | 0 | 110,000.00 Total Price 285,000.00 Total Discount 0.00 Net Total 285,000.00 ← Continue Shopping CHECK OUT →	P
4	Click on delete button in the shopping cart	Display confirm message and then success message	 Are you sure? Yes Cancel	P

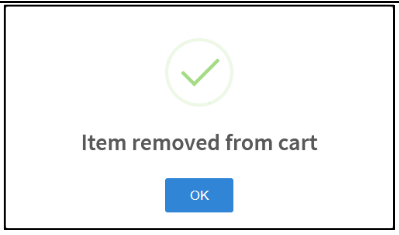
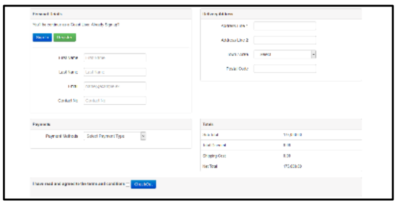
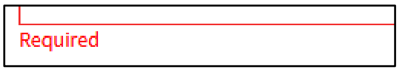
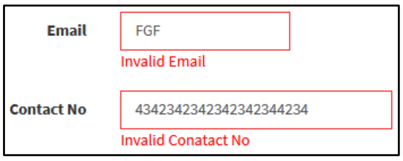
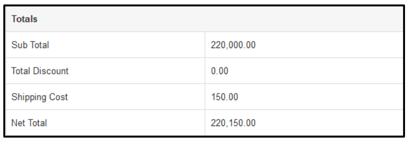
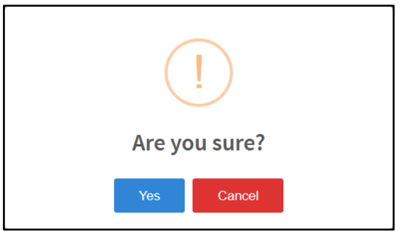
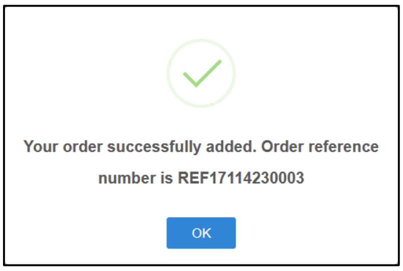
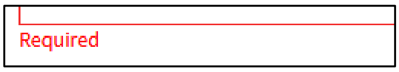
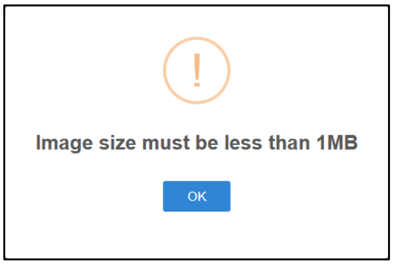
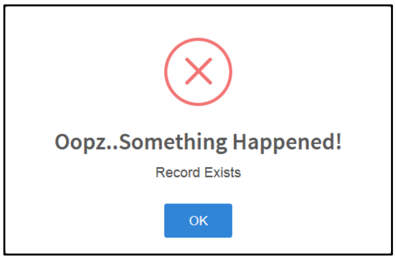
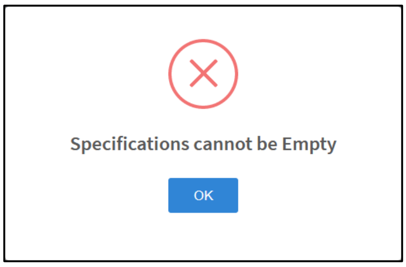
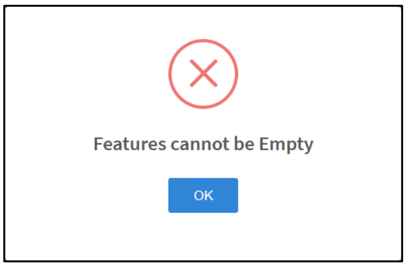
			 Item removed from cart OK	
5	Click on the CHECK OUT ➔	Redirect to checkout screen		P
6	Click on Confirm Checkout button with all fields empty	Display error	 Required message in all the required fields	P
7	Click on Confirm Checkout button with letters in email and invalid phone number	Display error	 Email: FGF Invalid Email Contact No: 4342342342342342344234 Invalid Conatact No	P
8	Change the quantity in the item list	Update the total	 Totals Sub Total: 220,000.00 Total Discount: 0.00 Shipping Cost: 150.00 Net Total: 220,150.00	P
9	Click on Confirm Checkout button with all valid fields	Display confirm message	 Are you sure? Yes Cancel	P
10	Confirm the message	Success message with reference number	 Your order successfully added. Order reference number is REF17114230003 OK	P

Table E.3: Test result for checkout process

Test case for the adding items can be shown in the Table E.4.

No	Test Case	Expected Output	Actual Output	P/F
1	Click on SAVE button with empty field	Display error	 Required message in all the required fields	P
2	Upload image with size greater than 1MB	Display error	 Image size must be less than 1MB 	P
3	Click on SAVE button with existing item code	Display error	 Oopz..Something Happened! Record Exists 	P
4	Click on SAVE button without adding specifications for the item	Display error	 Specifications cannot be Empty 	P
5	Click on SAVE button without adding features for the item	Display error	 Features cannot be Empty 	P

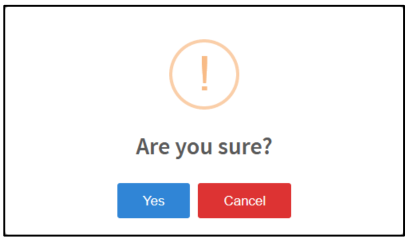
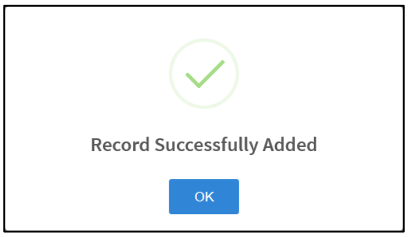
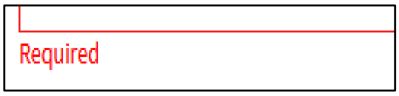
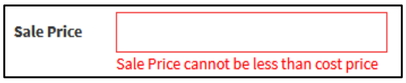
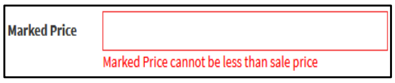
6	Click on SAVE button with all valid inputs	Display confirm message	 Are you sure? 	P
7	Confirm the message	Successful message display	 Record Successfully Added 	P

Table E.4: Test result for item add

Test case for the inventory assign can be shown in the Table E.5.

No	Test Case	Expected Output	Actual Output	P/F
1	Select item and search	Data loaded to grid	Data loaded to the grid which are only received	P
2	Click on SAVE button with empty field	Display error	 Required message in all the required fields	P
3	Click on SAVE button after filling sale price less than cost price	Display error	 Sale Price Sale Price cannot be less than cost price	P
4	Click on SAVE button after filling marked price less than sale price	Display error	 Marked Price Marked Price cannot be less than sale price	P

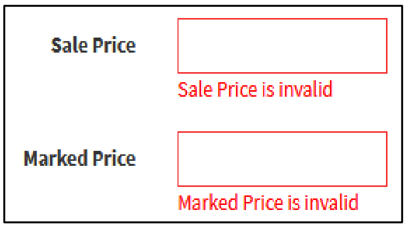
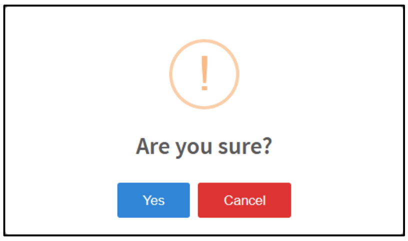
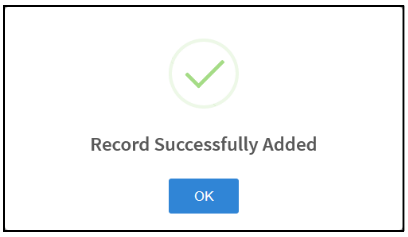
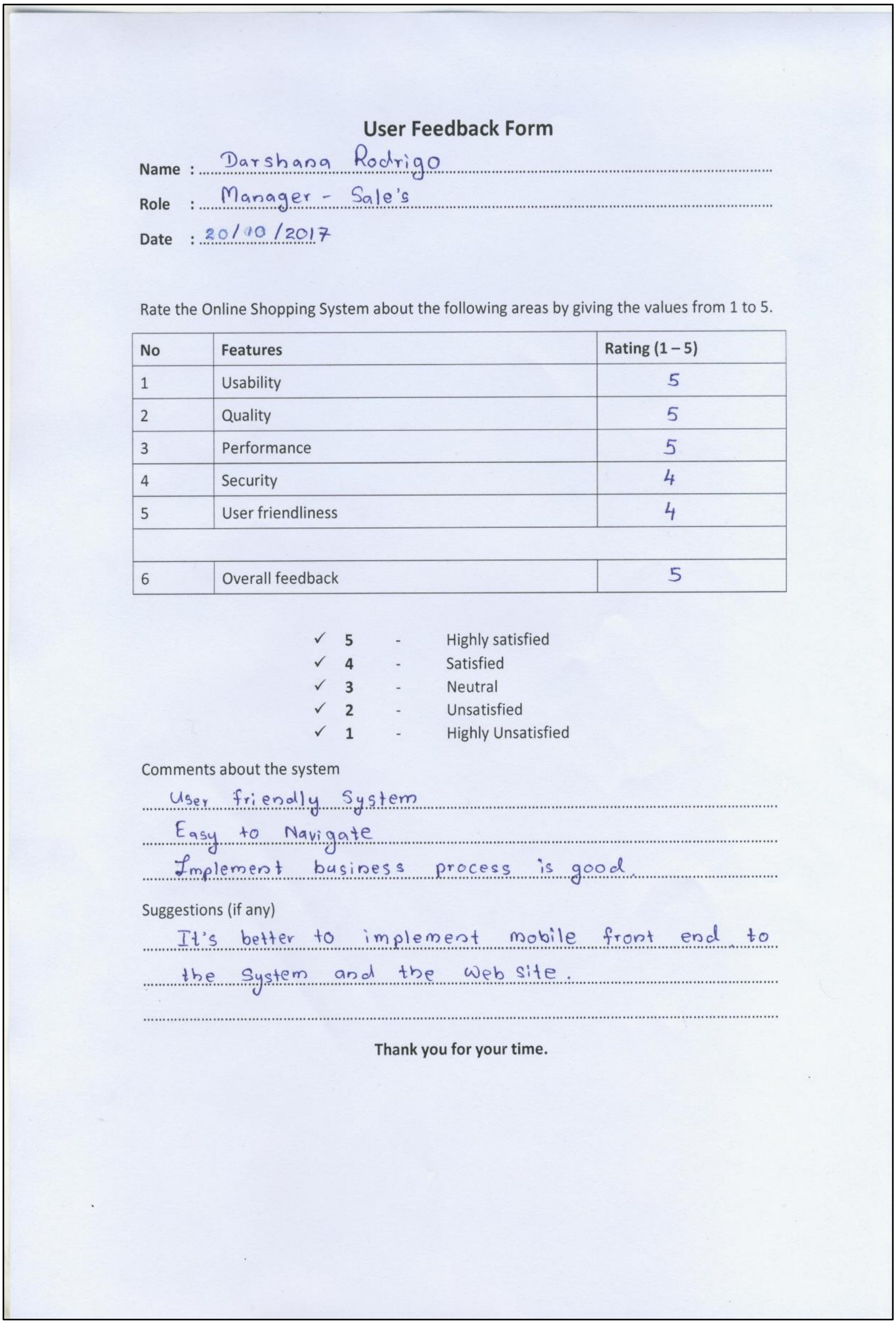
5	Click on SAVE button after filling fields with letters	Display error	 Sale Price Sale Price is invalid Marked Price Marked Price is invalid	P
6	Click on SAVE button with all valid inputs	Display confirm message	 Are you sure? 	P
7	Confirm the message	Successful message display	 Record Successfully Added 	P

Table E.5: Test result for inventory assign

Other than test cases, user feedback forms are also very important to evaluate the system. Chapter 5.4 described about that and the user feedback forms according to mention format in chapter 5 are shown below in the Figure E.1 and Figure E.2. Those form were collected from the users who work in the Ecart.lk (PVT) Ltd.



User Feedback Form

Name : Darshana Rodrigo

Role : Manager - Sale's

Date : 20/10/2017

Rate the Online Shopping System about the following areas by giving the values from 1 to 5.

No	Features	Rating (1 – 5)
1	Usability	5
2	Quality	5
3	Performance	5
4	Security	4
5	User friendliness	4
6	Overall feedback	5

- ✓ **5** - Highly satisfied
- ✓ **4** - Satisfied
- ✓ **3** - Neutral
- ✓ **2** - Unsatisfied
- ✓ **1** - Highly Unsatisfied

Comments about the system

User friendly System

Easy to Navigate

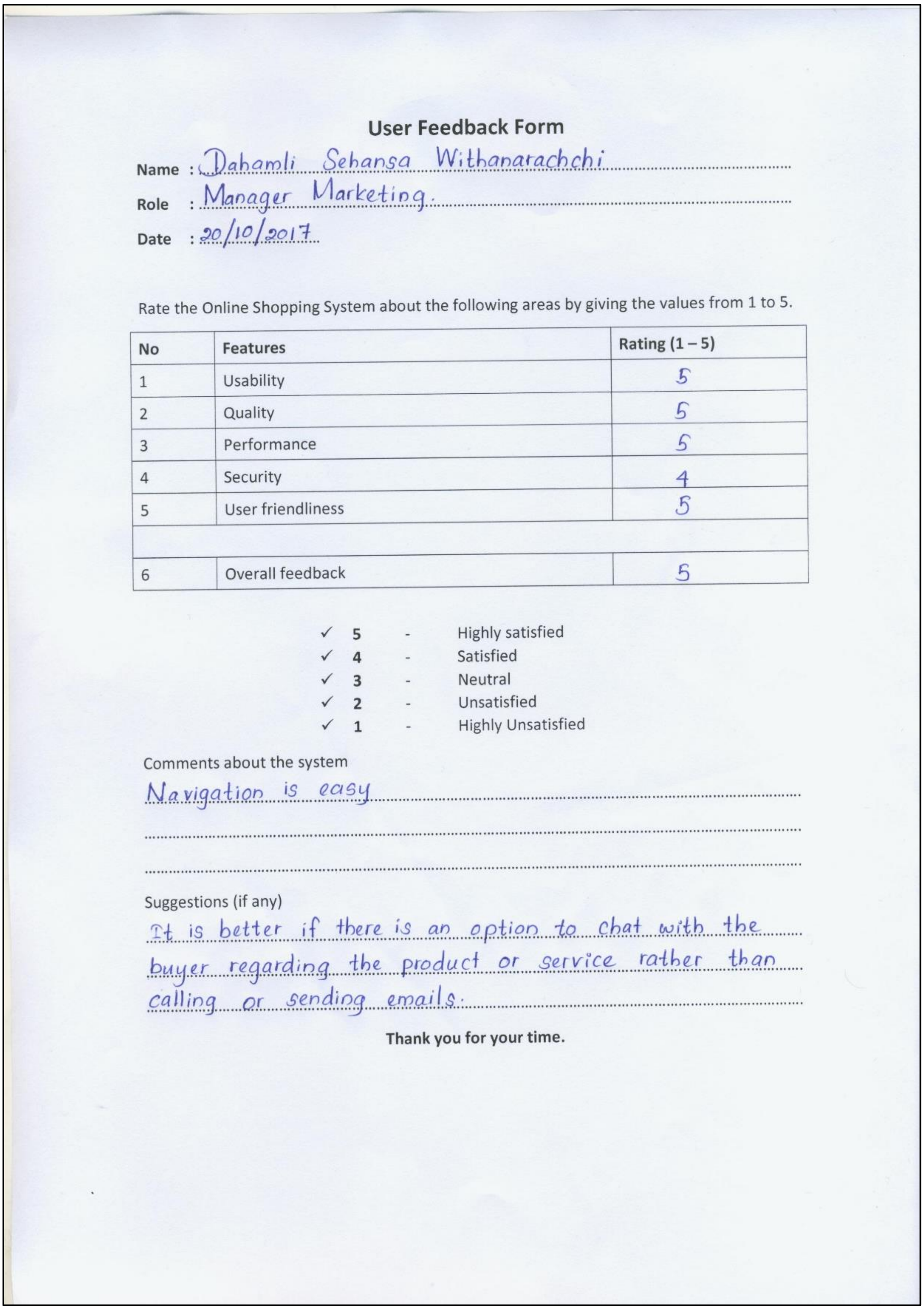
Implement business process is good.

Suggestions (if any)

It's better to implement mobile front end to the System and the web site.

Thank you for your time.

Figure E.1: User feedback form 1



User Feedback Form

Name : Dahamli Sehansa Withanarachchi

Role : Manager Marketing.

Date : 20/10/2017

Rate the Online Shopping System about the following areas by giving the values from 1 to 5.

No	Features	Rating (1 – 5)
1	Usability	5
2	Quality	5
3	Performance	5
4	Security	4
5	User friendliness	5
6	Overall feedback	5

✓	5	-	Highly satisfied
✓	4	-	Satisfied
✓	3	-	Neutral
✓	2	-	Unsatisfied
✓	1	-	Highly Unsatisfied

Comments about the system

Navigation is easy

Suggestions (if any)

It is better if there is an option to chat with the buyer regarding the product or service rather than calling or sending emails.

Thank you for your time.

Figure E.2: User feedback form 2

Appendix F - Code Listing

Appendix F is consist of major code segments which really important to the system. To represent all the code below figures shows the common code segments.

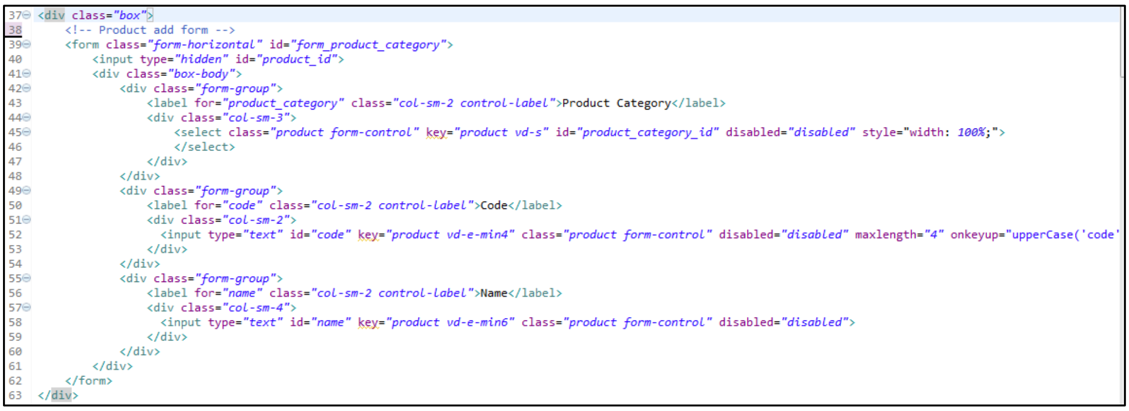
- Data Insert
- Data update
- Data delete
- Data view
- Report creation
- Excel Generation

These code segments can be found in the project source folder which is in the CD.

Data Insert

Following Figure F.1, Figure F.2, Figure F.3 and F.4 shows the data insert code segment of the adding product.

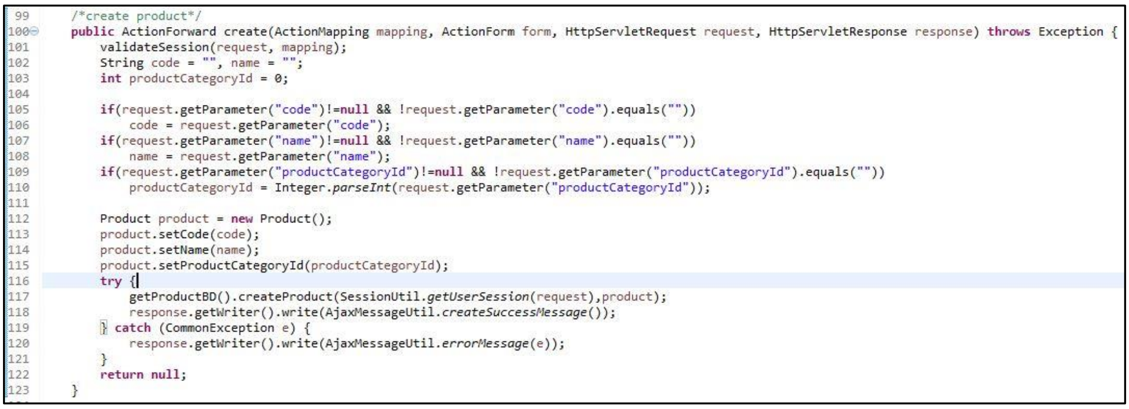
Product.jsp



```
<div class="box">
    <!-- Product add form -->
    <form class="form-horizontal" id="form_product_category">
        <input type="hidden" id="product_id">
        <div class="box-body">
            <div class="form-group">
                <label for="product_category" class="col-sm-2 control-label">Product Category</label>
                <div class="col-sm-3">
                    <select class="product form-control" key="product vd-s" id="product_category_id" disabled="disabled" style="width: 100%;">
                    </select>
                </div>
            </div>
            <div class="form-group">
                <label for="code" class="col-sm-2 control-label">Code</label>
                <div class="col-sm-2">
                  <input type="text" id="code" key="product vd-e-min4" class="product form-control" disabled="disabled" maxlength="4" onkeyup="upperCase('code'
                </div>
            </div>
            <div class="form-group">
                <label for="name" class="col-sm-2 control-label">Name</label>
                <div class="col-sm-4">
                  <input type="text" id="name" key="product vd-e-min6" class="product form-control" disabled="disabled">
                </div>
            </div>
        </div>
    </form>
</div>
```

Figure F.1: Form field of Product.jsp

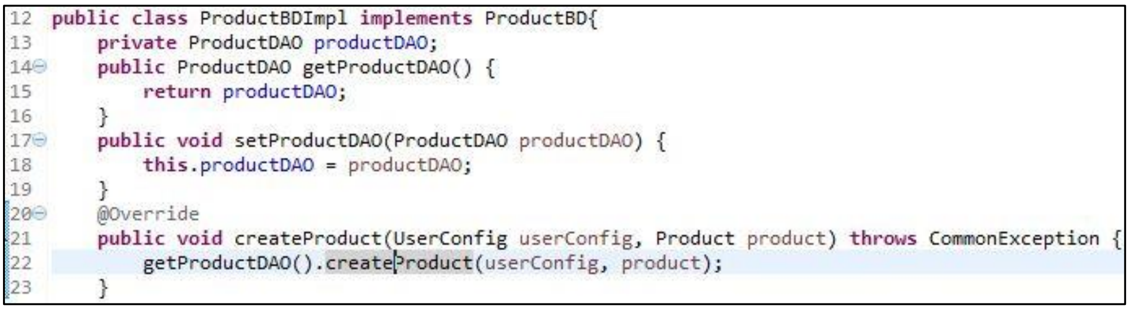
ProductAction.java



```
    /*create product*/
    public ActionForward create(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {
        validateSession(request, mapping);
        String code = "", name = "";
        int productCategoryId = 0;

        if(request.getParameter("code")!=null && !request.getParameter("code").equals(""))
            code = request.getParameter("code");
        if(request.getParameter("name")!=null && !request.getParameter("name").equals(""))
            name = request.getParameter("name");
        if(request.getParameter("productCategoryId")!=null && !request.getParameter("productCategoryId").equals(""))
            productCategoryId = Integer.parseInt(request.getParameter("productCategoryId"));

        Product product = new Product();
        product.setCode(code);
        product.setName(name);
        product.setProductCategoryId(productCategoryId);
        try {
            getProductBD().createProduct(SessionUtil.getUserSession(request),product);
            response.getWriter().write(AjaxMessageUtil.createSuccessMessage());
        } catch (CommonException e) {
            response.getWriter().write(AjaxMessageUtil.errorMessage(e));
        }
        return null;
    }
```

Figure F.2: Create method of ProductAction.java

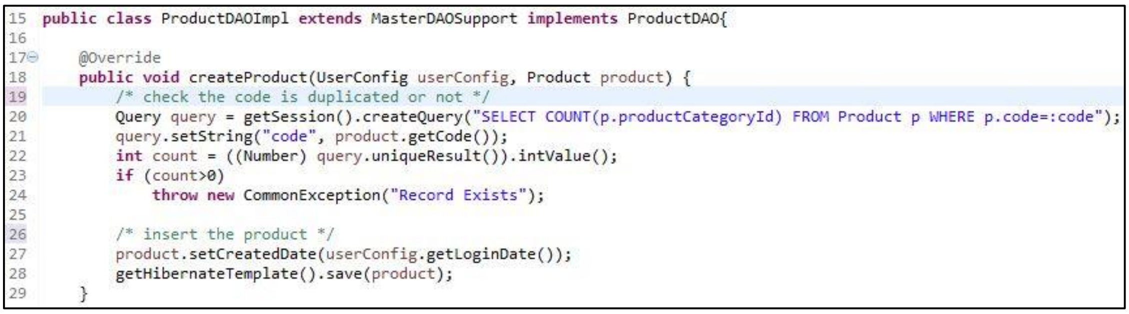
ProductBDImpl.java



```java
public class ProductBDImpl implements ProductBD{
    private ProductDAO productDAO;
    public ProductDAO getProductDAO() {
        return productDAO;
    }
    public void setProductDAO(ProductDAO productDAO) {
        this.productDAO = productDAO;
    }
    @Override
    public void createProduct(UserConfig userConfig, Product product) throws CommonException {
        getProductDAO().createProduct(userConfig, product);
    }
```

Figure F.3: Method implementation in ProductBDImpl.java

ProductDAOImpl.java



```java
public class ProductDAOImpl extends MasterDAOSupport implements ProductDAO{

    @Override
    public void createProduct(UserConfig userConfig, Product product) {
        /* check the code is duplicated or not */
        Query query = getSession().createQuery("SELECT COUNT(p.productCategoryId) FROM Product p WHERE p.code=:code");
        query.setString("code", product.getCode());
        int count = ((Number) query.uniqueResult()).intValue();
        if (count>0)
            throw new CommonException("Record Exists");

        /* insert the product */
        product.setCreatedDate(userConfig.getLoginDate());
        getHibernateTemplate().save(product);
    }
```

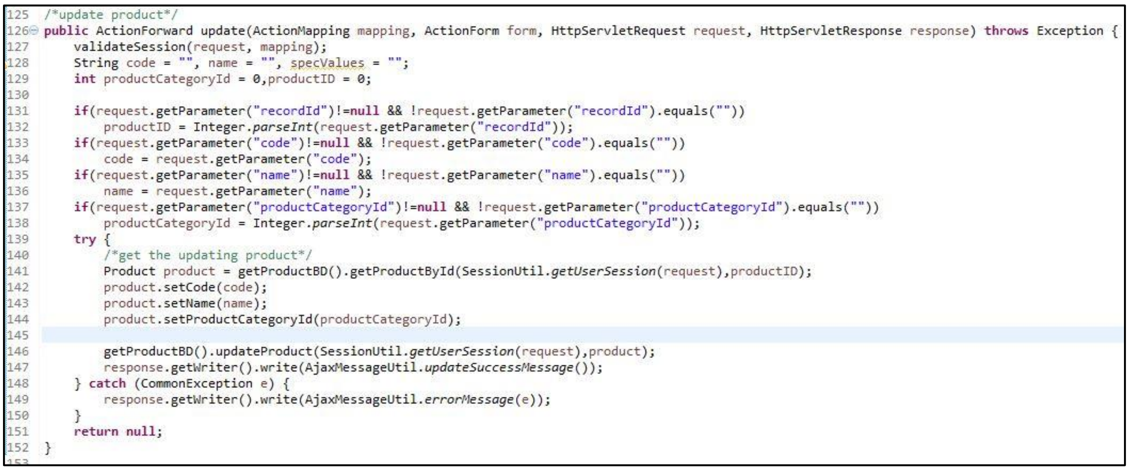
Figure F.4: Create method in ProductDAOImpl.java

Data update

Update code segment also can be describe through update product. That will be describe in the following Figures F.5, F.6 and F.7.

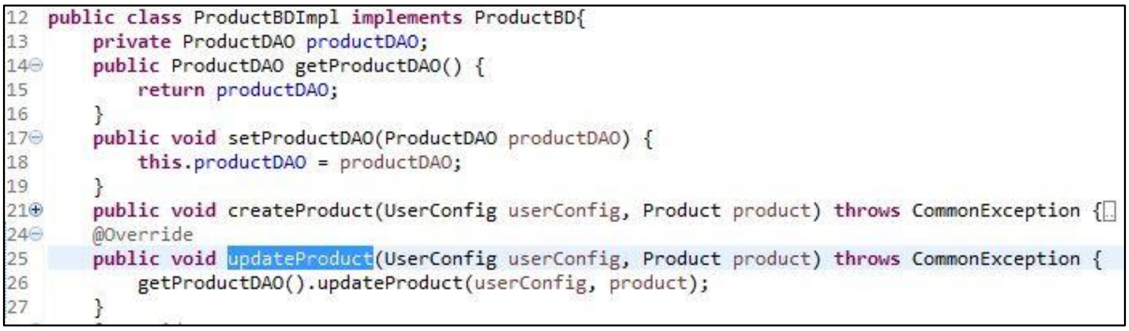
Product.jsp field form is same as the figure F.1.

ProductAction.java



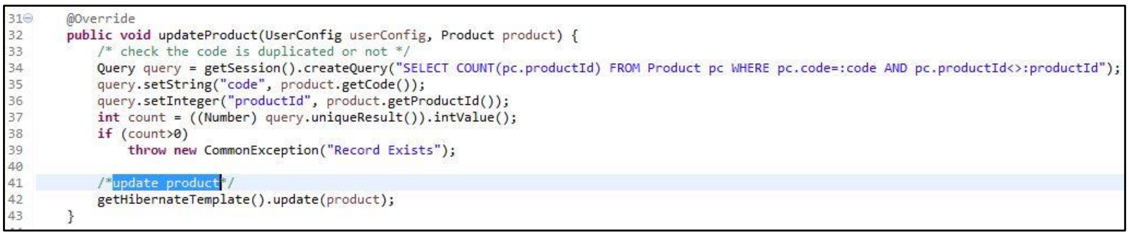
```java
/*update product*/
public ActionForward update(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {
    validateSession(request, mapping);
    String code = "", name = "", specValues = "";
    int productCategoryId = 0,productID = 0;

    if(request.getParameter("recordId")!=null && !request.getParameter("recordId").equals(""))
        productID = Integer.parseInt(request.getParameter("recordId"));
    if(request.getParameter("code")!=null && !request.getParameter("code").equals(""))
        code = request.getParameter("code");
    if(request.getParameter("name")!=null && !request.getParameter("name").equals(""))
        name = request.getParameter("name");
    if(request.getParameter("productCategoryId")!=null && !request.getParameter("productCategoryId").equals(""))
        productCategoryId = Integer.parseInt(request.getParameter("productCategoryId"));
    try {
        /*get the updating product*/
        Product product = getProductBD().getProductById(SessionUtil.getUserSession(request),productID);
        product.setCode(code);
        product.setName(name);
        product.setProductCategoryId(productCategoryId);

        getProductBD().updateProduct(SessionUtil.getUserSession(request),product);
        response.getWriter().write(AjaxMessageUtil.updateSuccessMessage());
    } catch (CommonException e) {
        response.getWriter().write(AjaxMessageUtil.errorMessage(e));
    }
    return null;
}
```

Figure F.5: Update method of the ProductAction.java



```java
public class ProductBDImpl implements ProductBD{
    private ProductDAO productDAO;
    public ProductDAO getProductDAO() {
        return productDAO;
    }
    public void setProductDAO(ProductDAO productDAO) {
        this.productDAO = productDAO;
    }
    public void createProduct(UserConfig userConfig, Product product) throws CommonException {□
    @Override
    public void updateProduct(UserConfig userConfig, Product product) throws CommonException {
        getProductDAO().updateProduct(userConfig, product);
    }
```

Figure F.6: Update method ProductBDImpl.java

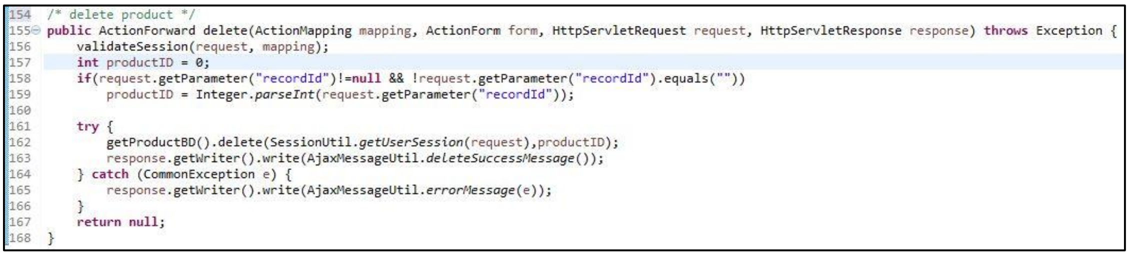


```java
    @Override
    public void updateProduct(UserConfig userConfig, Product product) {
        /* check the code is duplicated or not */
        Query query = getSession().createQuery("SELECT COUNT(pc.productId) FROM Product pc WHERE pc.code=:code AND pc.productId<>:productId");
        query.setString("code", product.getCode());
        query.setInteger("productId", product.getProductId());
        int count = ((Number) query.uniqueResult()).intValue();
        if (count>0)
            throw new CommonException("Record Exists");

        /*update product*/
        getHibernateTemplate().update(product);
    }
```

Figure F.7: Update method of ProductDAOImpl.java

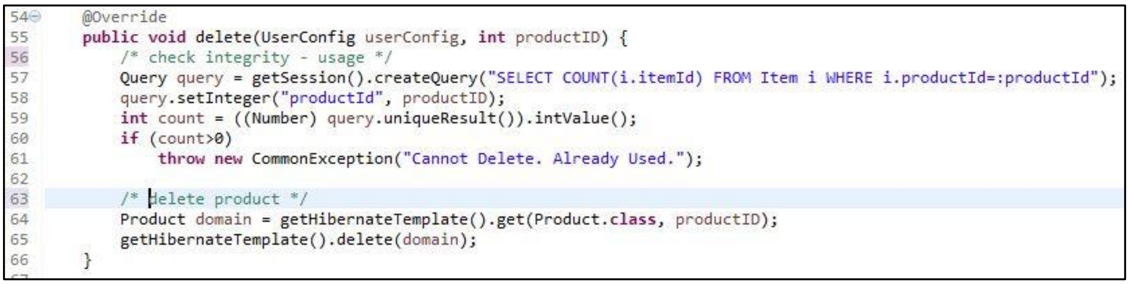
Data Delete

Figure F.8 and F.9 demonstrate the delete methods of the system.



```java
/* delete product */
public ActionForward delete(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {
    validateSession(request, mapping);
    int productID = 0;
    if(request.getParameter("recordId")!=null && !request.getParameter("recordId").equals(""))
        productID = Integer.parseInt(request.getParameter("recordId"));

    try {
        getProductBD().delete(SessionUtil.getUserSession(request),productID);
        response.getWriter().write(AjaxMessageUtil.deleteSuccessMessage());
    } catch (CommonException e) {
        response.getWriter().write(AjaxMessageUtil.errorMessage(e));
    }
    return null;
}
```

Figure F.8: Delete method of ProductAction.java

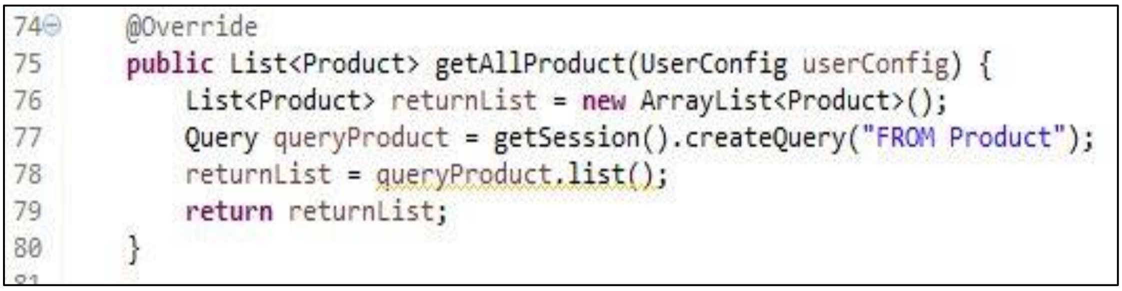


```java
    @Override
    public void delete(UserConfig userConfig, int productID) {
        /* check integrity - usage */
        Query query = getSession().createQuery("SELECT COUNT(i.itemId) FROM Item i WHERE i.productId=:productId");
        query.setInteger("productId", productID);
        int count = ((Number) query.uniqueResult()).intValue();
        if (count>0)
            throw new CommonException("Cannot Delete. Already Used.");

        /* delete product */
        Product domain = getHibernateTemplate().get(Product.class, productID);
        getHibernateTemplate().delete(domain);
    }
```

Figure F.9: Delete method of ProductDAOImpl.java

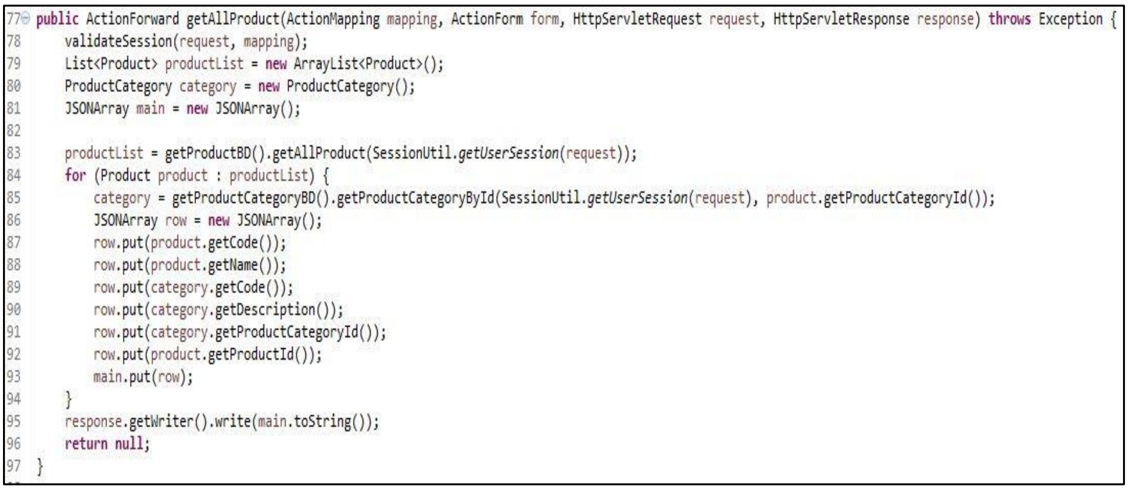
Data View

Saved data in the database can be view in the data grid. Figure F.10 represent the data retrieve method in the ProductDAOImpl.java, Figure F.11 shows how it manipulate and Figure F.12 shows the how retrieved data represent in the data grid.



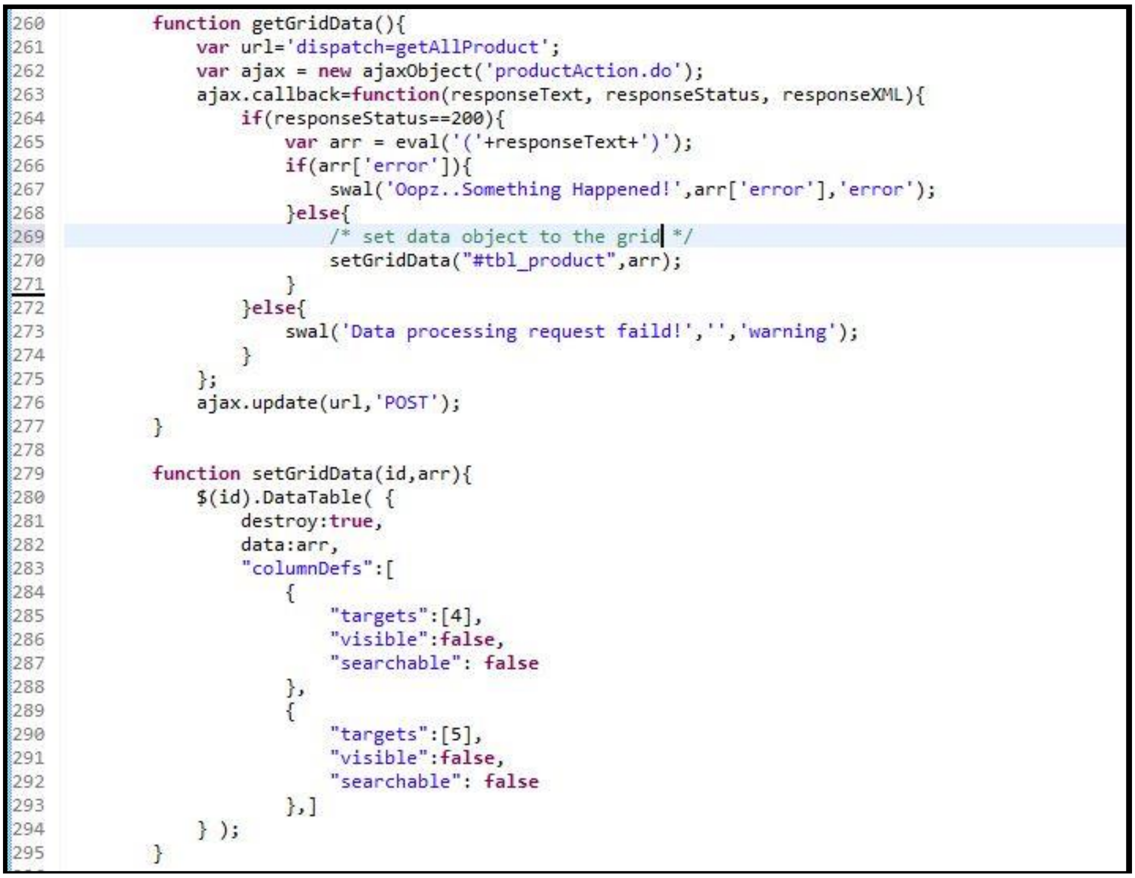
```java
    @Override
    public List<Product> getAllProduct(UserConfig userConfig) {
        List<Product> returnList = new ArrayList<Product>();
        Query queryProduct = getSession().createQuery("FROM Product");
        returnList = queryProduct.list();
        return returnList;
    }
```

Figure F.10: Get data method in ProductDAOImpl.java



```java
public ActionForward getAllProduct(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {
    validateSession(request, mapping);
    List<Product> productList = new ArrayList<Product>();
    ProductCategory category = new ProductCategory();
    JSONArray main = new JSONArray();

    productList = getProductBD().getAllProduct(SessionUtil.getUserSession(request));
    for (Product product : productList) {
        category = getProductCategoryBD().getProductCategoryById(SessionUtil.getUserSession(request), product.getProductCategoryId());
        JSONArray row = new JSONArray();
        row.put(product.getCode());
        row.put(product.getName());
        row.put(category.getCode());
        row.put(category.getDescription());
        row.put(category.getProductCategoryId());
        row.put(product.getProductId());
        main.put(row);
    }
    response.getWriter().write(main.toString());
    return null;
}
```

Figure F.11: Data manipulate in ProductAction.java

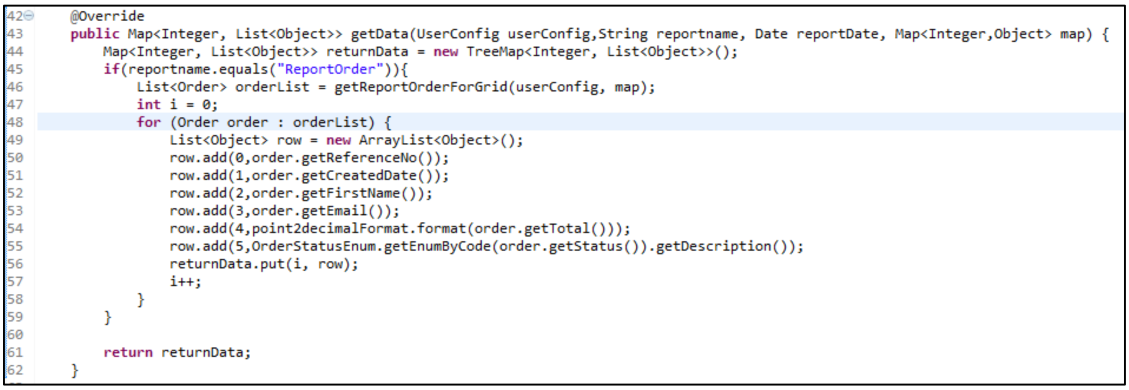


```
        function getGridData(){
            var url='dispatch=getAllProduct';
            var ajax = new ajaxObject('productAction.do');
            ajax.callback=function(responseText, responseStatus, responseXML){
                if(responseStatus==200){
                    var arr = eval('('+responseText+')');
                    if(arr['error']){
                        swal('Oopz..Something Happened!',arr['error'],'error');
                    }else{
                        /* set data object to the grid */
                        setGridData("#tbl_product",arr);
                    }
                }else{
                    swal('Data processing request faild!','','warning');
                }
            };
            ajax.update(url,'POST');
        }

        function setGridData(id,arr){
            $(id).DataTable( {
                destroy:true,
                data:arr,
                "columnDefs":[
                    {
                        "targets":[4],
                        "visible":false,
                        "searchable": false
                    },
                    {
                        "targets":[5],
                        "visible":false,
                        "searchable": false
                    },]
            } );
        }
```

Figure F.12: Data set into data grid in Product.jsp

Report Creation

Sample code segment for the report generation of the order detail is shown in the Figure F.13.

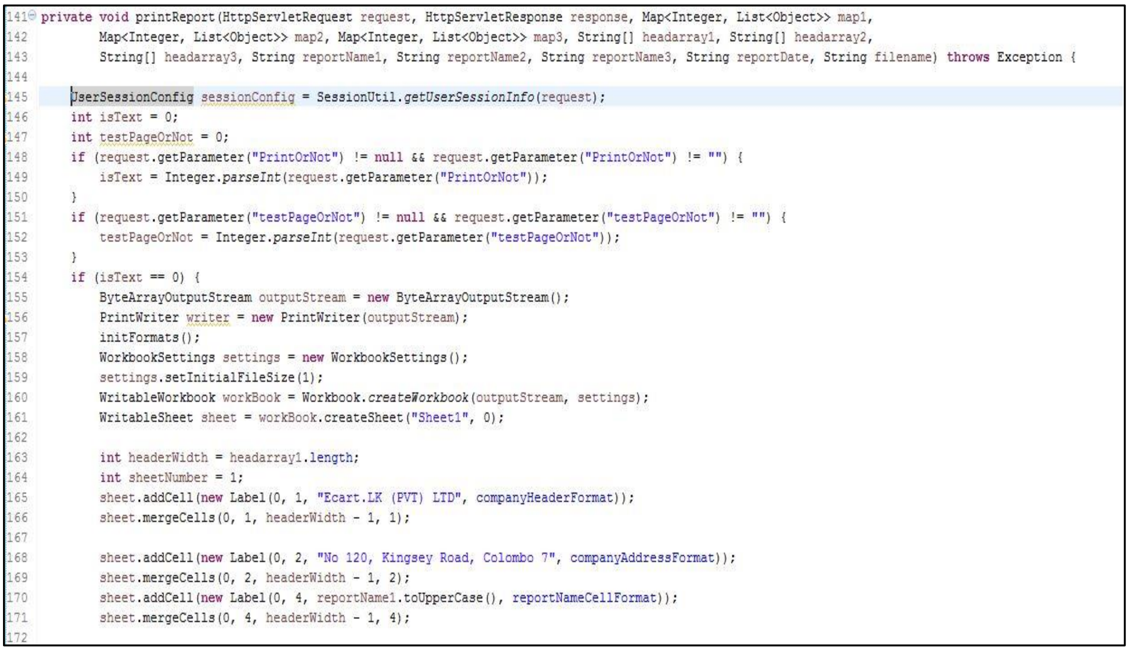


```
    @Override
    public Map<Integer, List<Object>> getData(UserConfig userConfig,String reportname, Date reportDate, Map<Integer,Object> map) {
        Map<Integer, List<Object>> returnData = new TreeMap<Integer, List<Object>>();
        if(reportname.equals("ReportOrder")){
            List<Order> orderList = getReportOrderForGrid(userConfig, map);
            int i = 0;
            for (Order order : orderList) {
                List<Object> row = new ArrayList<Object>();
                row.add(0,order.getReferenceNo());
                row.add(1,order.getCreatedDate());
                row.add(2,order.getFirstName());
                row.add(3,order.getEmail());
                row.add(4,point2decimalFormat.format(order.getTotal()));
                row.add(5,OrderStatusEnum.getEnumByCode(order.getStatus()).getDescription());
                returnData.put(i, row);
                i++;
            }
        }

        return returnData;
    }
```

Figure F.13: Order detail report creation

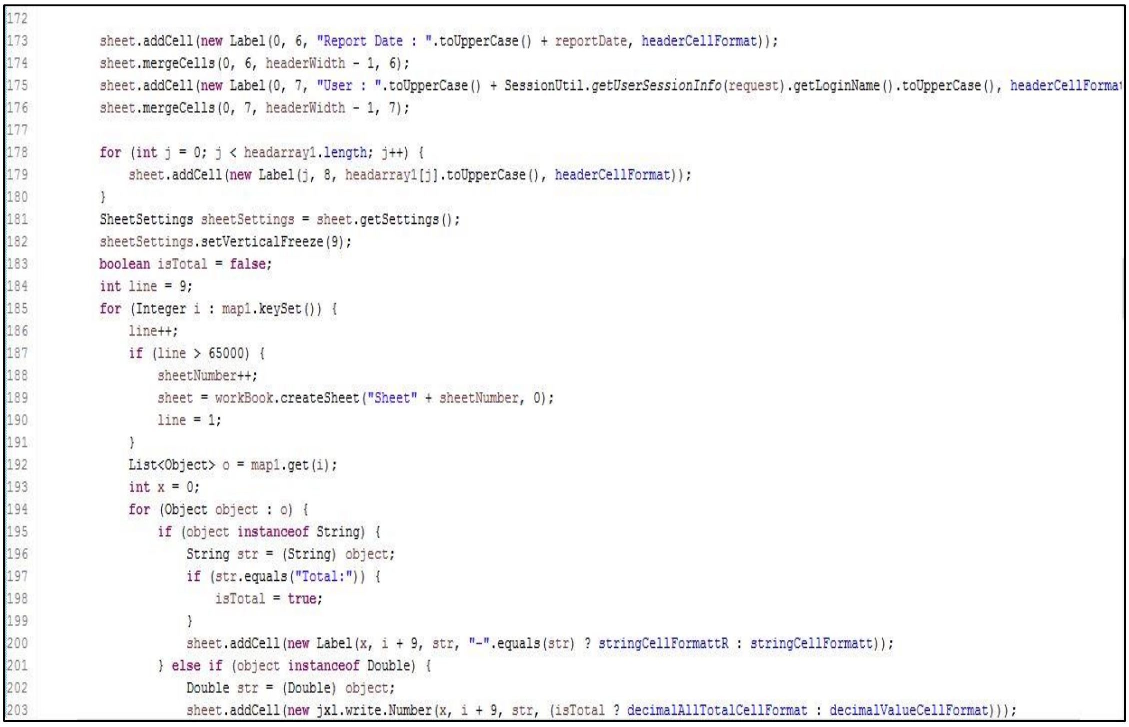
Excel Generation

Excel generation common function was created by the jxl library and common method is shown in below Figures F.14, F.15, F.16, F.17, F.18 and F.19.



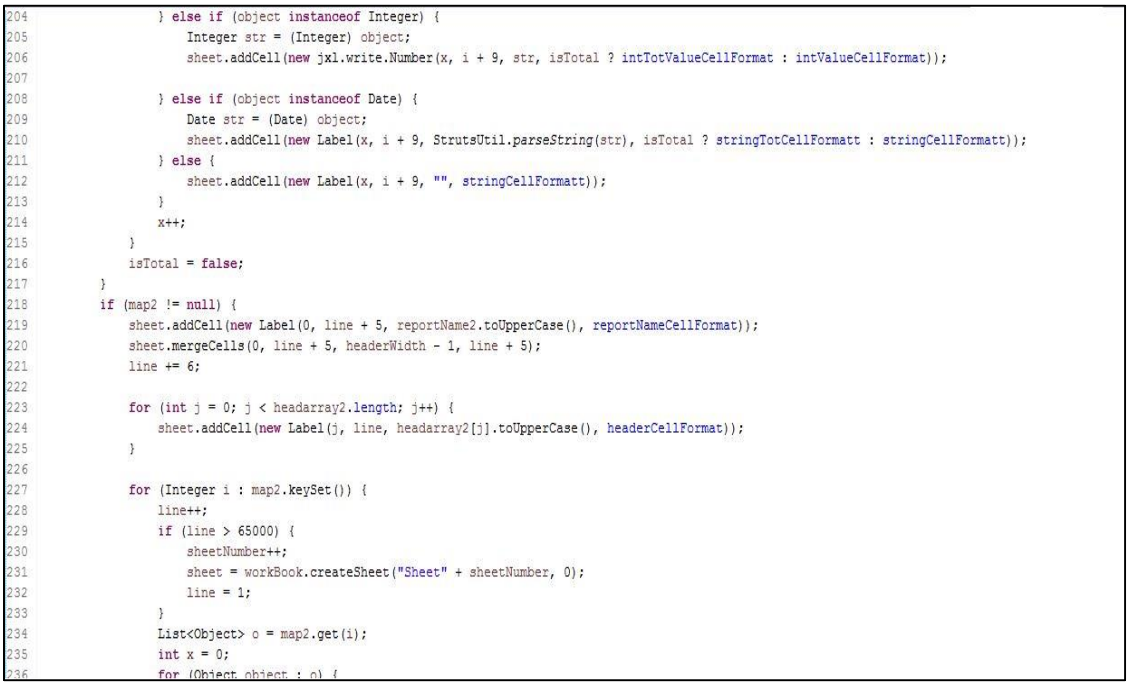
```java
private void printReport(HttpServletRequest request, HttpServletResponse response, Map<Integer, List<Object>> map1,
        Map<Integer, List<Object>> map2, Map<Integer, List<Object>> map3, String[] headarray1, String[] headarray2,
        String[] headarray3, String reportName1, String reportName2, String reportName3, String reportDate, String filename) throws Exception {

    UserSessionConfig sessionConfig = SessionUtil.getUserSessionInfo(request);
    int isText = 0;
    int testPageOrNot = 0;
    if (request.getParameter("PrintOrNot") != null && request.getParameter("PrintOrNot") != "") {
        isText = Integer.parseInt(request.getParameter("PrintOrNot"));
    }
    if (request.getParameter("testPageOrNot") != null && request.getParameter("testPageOrNot") != "") {
        testPageOrNot = Integer.parseInt(request.getParameter("testPageOrNot"));
    }
    if (isText == 0) {
        ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
        PrintWriter writer = new PrintWriter(outputStream);
        initFormats();
        WorkbookSettings settings = new WorkbookSettings();
        settings.setInitialFileSize(1);
        WritableWorkbook workBook = Workbook.createWorkbook(outputStream, settings);
        WritableSheet sheet = workBook.createSheet("Sheet1", 0);

        int headerWidth = headarray1.length;
        int sheetNumber = 1;
        sheet.addCell(new Label(0, 1, "Ecart.LK (PVT) LTD", companyHeaderFormat));
        sheet.mergeCells(0, 1, headerWidth - 1, 1);

        sheet.addCell(new Label(0, 2, "No 120, Kingsey Road, Colombo 7", companyAddressFormat));
        sheet.mergeCells(0, 2, headerWidth - 1, 2);
        sheet.addCell(new Label(0, 4, reportName1.toUpperCase(), reportNameCellFormat));
        sheet.mergeCells(0, 4, headerWidth - 1, 4);

```

Figure F.14: Excel generation code sample part 1



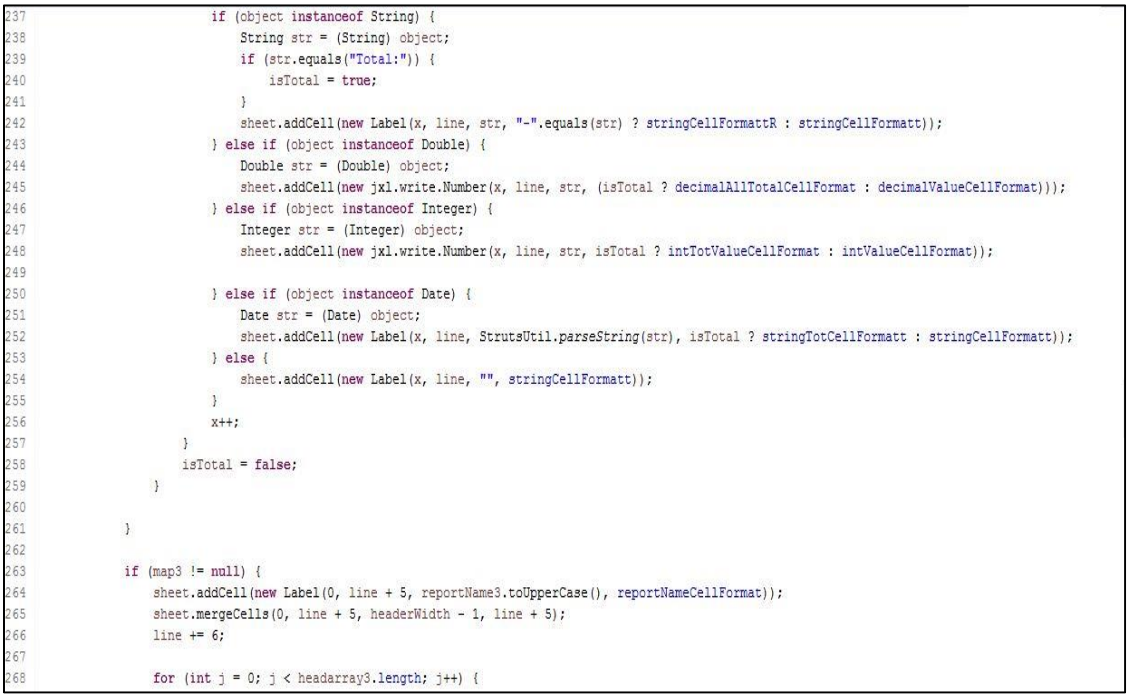
```java

        sheet.addCell(new Label(0, 6, "Report Date : ".toUpperCase() + reportDate, headerCellFormat));
        sheet.mergeCells(0, 6, headerWidth - 1, 6);
        sheet.addCell(new Label(0, 7, "User : ".toUpperCase() + SessionUtil.getUserSessionInfo(request).getLoginName().toUpperCase(), headerCellForma
        sheet.mergeCells(0, 7, headerWidth - 1, 7);

        for (int j = 0; j < headarray1.length; j++) {
            sheet.addCell(new Label(j, 8, headarray1[j].toUpperCase(), headerCellFormat));
        }
        SheetSettings sheetSettings = sheet.getSettings();
        sheetSettings.setVerticalFreeze(9);
        boolean isTotal = false;
        int line = 9;
        for (Integer i : map1.keySet()) {
            line++;
            if (line > 65000) {
                sheetNumber++;
                sheet = workBook.createSheet("Sheet" + sheetNumber, 0);
                line = 1;
            }
            List<Object> o = map1.get(i);
            int x = 0;
            for (Object object : o) {
                if (object instanceof String) {
                    String str = (String) object;
                    if (str.equals("Total:")) {
                        isTotal = true;
                    }
                    sheet.addCell(new Label(x, i + 9, str, "-".equals(str) ? stringCellFormattR : stringCellFormatt));
                } else if (object instanceof Double) {
                    Double str = (Double) object;
                    sheet.addCell(new jxl.write.Number(x, i + 9, str, (isTotal ? decimalAllTotalCellFormat : decimalValueCellFormat)));
```

Figure F.15: Excel generation code sample part 2



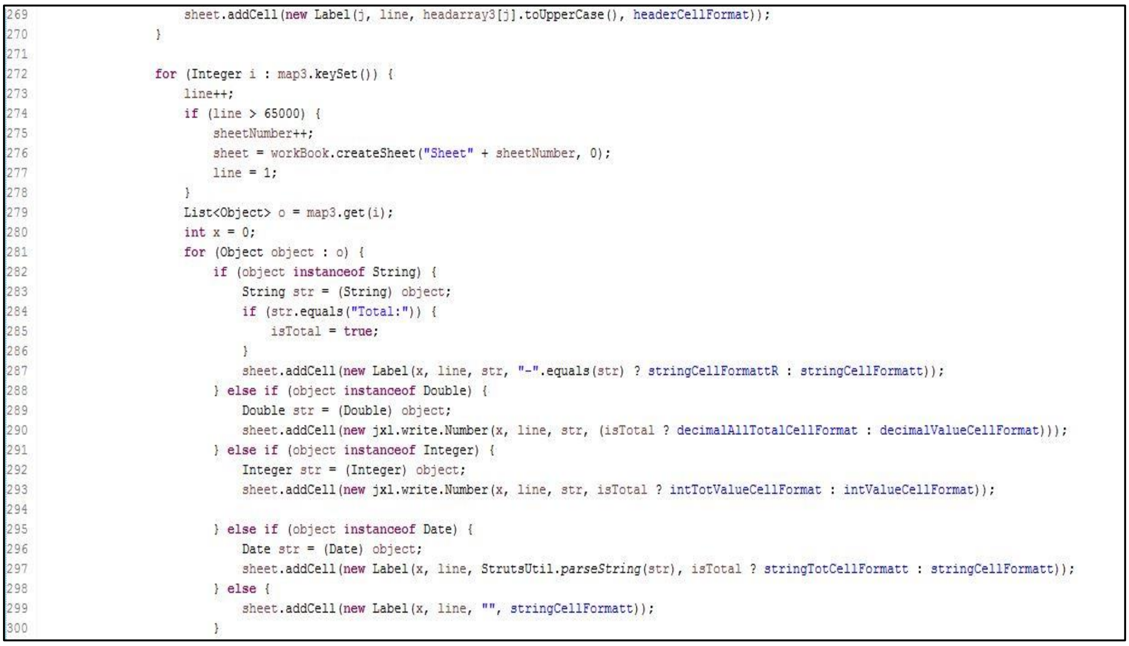
```
                } else if (object instanceof Integer) {
                    Integer str = (Integer) object;
                    sheet.addCell(new jxl.write.Number(x, i + 9, str, isTotal ? intTotValueCellFormat : intValueCellFormat));

                } else if (object instanceof Date) {
                    Date str = (Date) object;
                    sheet.addCell(new Label(x, i + 9, StrutsUtil.parseString(str), isTotal ? stringTotCellFormatt : stringCellFormatt));
                } else {
                    sheet.addCell(new Label(x, i + 9, "", stringCellFormatt));
                }
                x++;
            }
            isTotal = false;
        }
        if (map2 != null) {
            sheet.addCell(new Label(0, line + 5, reportName2.toUpperCase(), reportNameCellFormat));
            sheet.mergeCells(0, line + 5, headerWidth - 1, line + 5);
            line += 6;

            for (int j = 0; j < headarray2.length; j++) {
                sheet.addCell(new Label(j, line, headarray2[j].toUpperCase(), headerCellFormat));
            }

            for (Integer i : map2.keySet()) {
                line++;
                if (line > 65000) {
                    sheetNumber++;
                    sheet = workBook.createSheet("Sheet" + sheetNumber, 0);
                    line = 1;
                }
                List<Object> o = map2.get(i);
                int x = 0;
                for (Object object : o) {
```

Figure F.16: Excel generation code sample part 3



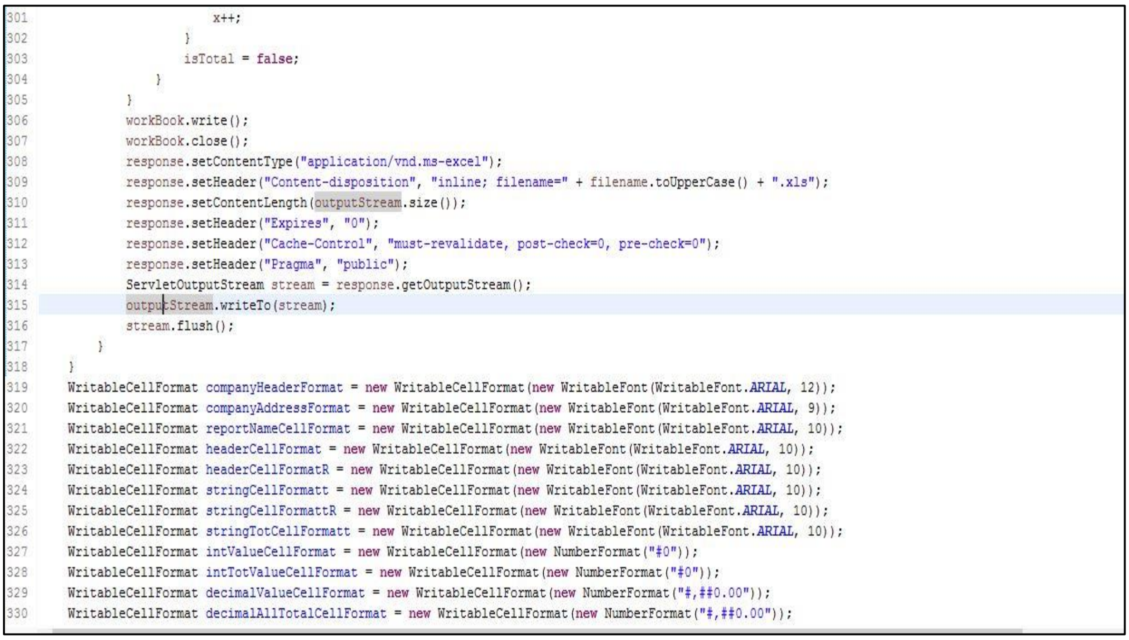
```
                    if (object instanceof String) {
                        String str = (String) object;
                        if (str.equals("Total:")) {
                            isTotal = true;
                        }
                        sheet.addCell(new Label(x, line, str, "-".equals(str) ? stringCellFormattR : stringCellFormatt));
                    } else if (object instanceof Double) {
                        Double str = (Double) object;
                        sheet.addCell(new jxl.write.Number(x, line, str, (isTotal ? decimalAllTotalCellFormat : decimalValueCellFormat)));
                    } else if (object instanceof Integer) {
                        Integer str = (Integer) object;
                        sheet.addCell(new jxl.write.Number(x, line, str, isTotal ? intTotValueCellFormat : intValueCellFormat));

                    } else if (object instanceof Date) {
                        Date str = (Date) object;
                        sheet.addCell(new Label(x, line, StrutsUtil.parseString(str), isTotal ? stringTotCellFormatt : stringCellFormatt));
                    } else {
                        sheet.addCell(new Label(x, line, "", stringCellFormatt));
                    }
                    x++;
                }
                isTotal = false;
            }

        }

        if (map3 != null) {
            sheet.addCell(new Label(0, line + 5, reportName3.toUpperCase(), reportNameCellFormat));
            sheet.mergeCells(0, line + 5, headerWidth - 1, line + 5);
            line += 6;

            for (int j = 0; j < headarray3.length; j++) {
```

Figure F.17: Excel generation code sample part 4



```java
                    sheet.addCell(new Label(j, line, headarray3[j].toUpperCase(), headerCellFormat));
                }

                for (Integer i : map3.keySet()) {
                    line++;
                    if (line > 65000) {
                        sheetNumber++;
                        sheet = workBook.createSheet("Sheet" + sheetNumber, 0);
                        line = 1;
                    }
                    List<Object> o = map3.get(i);
                    int x = 0;
                    for (Object object : o) {
                        if (object instanceof String) {
                            String str = (String) object;
                            if (str.equals("Total:")) {
                                isTotal = true;
                            }
                            sheet.addCell(new Label(x, line, str, "-".equals(str) ? stringCellFormattR : stringCellFormatt));
                        } else if (object instanceof Double) {
                            Double str = (Double) object;
                            sheet.addCell(new jxl.write.Number(x, line, str, (isTotal ? decimalAllTotalCellFormat : decimalValueCellFormat)));
                        } else if (object instanceof Integer) {
                            Integer str = (Integer) object;
                            sheet.addCell(new jxl.write.Number(x, line, str, isTotal ? intTotValueCellFormat : intValueCellFormat));

                        } else if (object instanceof Date) {
                            Date str = (Date) object;
                            sheet.addCell(new Label(x, line, StrutsUtil.parseString(str), isTotal ? stringTotCellFormatt : stringCellFormatt));
                        } else {
                            sheet.addCell(new Label(x, line, "", stringCellFormatt));
                        }
```

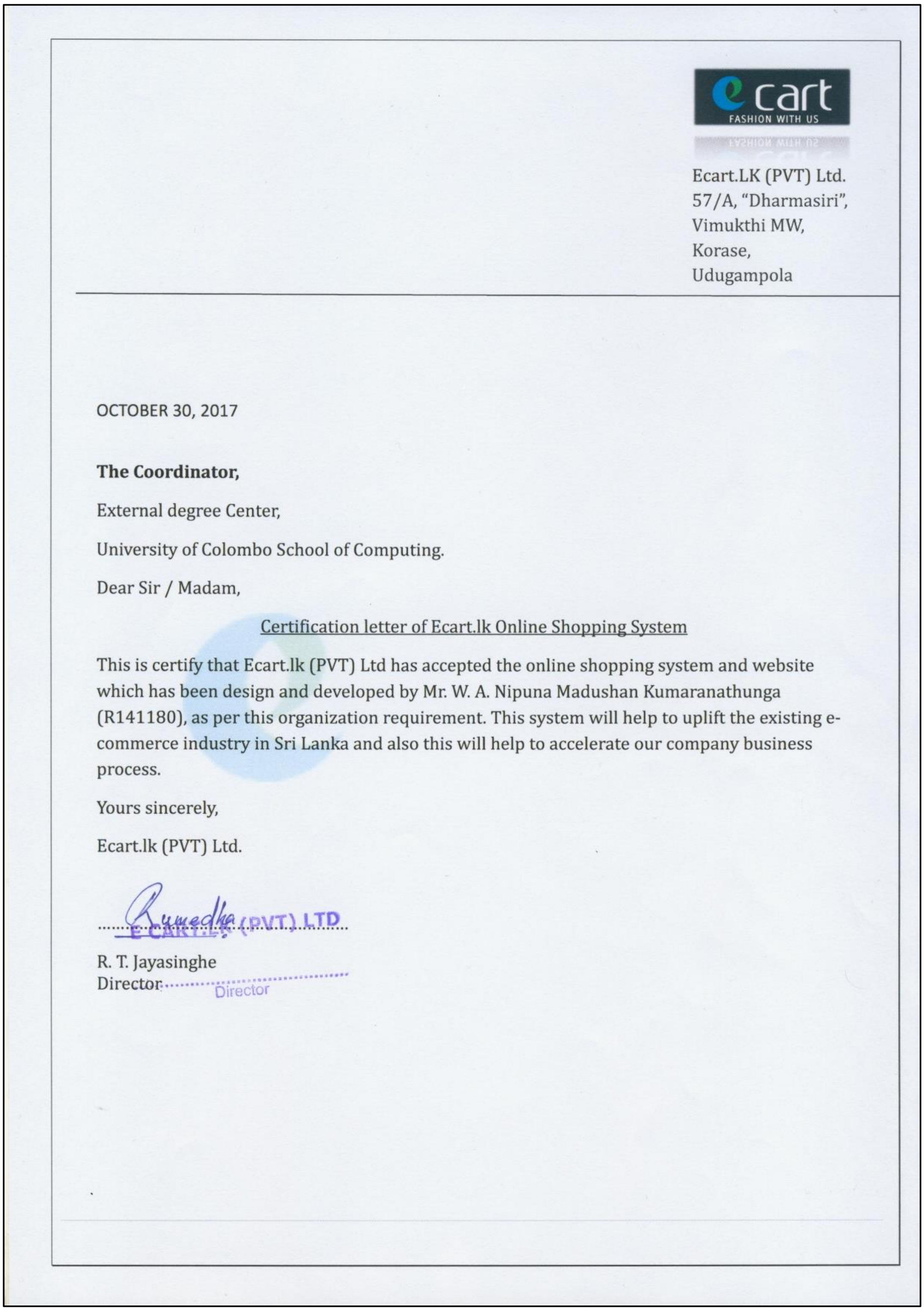
Figure F.18: Excel generation code sample part 5



```java
                        x++;
                    }
                    isTotal = false;
                }
            }
            workBook.write();
            workBook.close();
            response.setContentType("application/vnd.ms-excel");
            response.setHeader("Content-disposition", "inline; filename=" + filename.toUpperCase() + ".xls");
            response.setContentLength(outputStream.size());
            response.setHeader("Expires", "0");
            response.setHeader("Cache-Control", "must-revalidate, post-check=0, pre-check=0");
            response.setHeader("Pragma", "public");
            ServletOutputStream stream = response.getOutputStream();
            outputStream.writeTo(stream);
            stream.flush();
        }
    }
    WritableCellFormat companyHeaderFormat = new WritableCellFormat(new WritableFont(WritableFont.ARIAL, 12));
    WritableCellFormat companyAddressFormat = new WritableCellFormat(new WritableFont(WritableFont.ARIAL, 9));
    WritableCellFormat reportNameCellFormat = new WritableCellFormat(new WritableFont(WritableFont.ARIAL, 10));
    WritableCellFormat headerCellFormat = new WritableCellFormat(new WritableFont(WritableFont.ARIAL, 10));
    WritableCellFormat headerCellFormatR = new WritableCellFormat(new WritableFont(WritableFont.ARIAL, 10));
    WritableCellFormat stringCellFormatt = new WritableCellFormat(new WritableFont(WritableFont.ARIAL, 10));
    WritableCellFormat stringCellFormattR = new WritableCellFormat(new WritableFont(WritableFont.ARIAL, 10));
    WritableCellFormat stringTotCellFormatt = new WritableCellFormat(new WritableFont(WritableFont.ARIAL, 10));
    WritableCellFormat intValueCellFormat = new WritableCellFormat(new NumberFormat("#0"));
    WritableCellFormat intTotValueCellFormat = new WritableCellFormat(new NumberFormat("#0"));
    WritableCellFormat decimalValueCellFormat = new WritableCellFormat(new NumberFormat("#,##0.00"));
    WritableCellFormat decimalAllTotalCellFormat = new WritableCellFormat(new NumberFormat("#,##0.00"));
```

Figure F.19: Excel generation code sample part 6

Appendix G - Client Certificate



e cart
FASHION WITH US

Ecart.LK (PVT) Ltd.
57/A, "Dharmasiri",
Vimukthi MW,
Korase,
Udugampola

OCTOBER 30, 2017

The Coordinator,

External degree Center,

University of Colombo School of Computing.

Dear Sir / Madam,

Certification letter of Ecart.lk Online Shopping System

This is certify that Ecart.lk (PVT) Ltd has accepted the online shopping system and website which has been design and developed by Mr. W. A. Nipuna Madushan Kumaranathunga (R141180), as per this organization requirement. This system will help to uplift the existing e-commerce industry in Sri Lanka and also this will help to accelerate our company business process.

Yours sincerely,

Ecart.lk (PVT) Ltd.

Rumedha

E CART.LK (PVT) LTD

R. T. Jayasinghe
Director

Director

Glossary

Actor – is a stakeholder, organization or external system who has an interactions with your system. Actors are drawn as stick figures use case diagram.

Architecture – The design or structure of any kind of system.

Class – A class contains of a group of types of encapsulated instance variables and types of methods.

CSS – This is a language that is used for make attractive a content of the web pages.

Entity – A single object which can be store data.

JSP – Server side programming language based on java which stand for Java Server Pages.

Module – Separate component of a system which are interact with each other

Object Oriented – A software design method that models the real objects using classes and objects.

Server – A computer or computer program. It can manages access to a centralized resource in a network

UML – is a modeling language which can be used to visualize a design of the system.

XML – It is a Meta language that allow user to define their own user defined marked up language.

Index